metano – An open-source toolbox for the reconstruction and analysis of genome-scale metabolic models

Von der Fakultät für Lebenswissenschaften

der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

genehmigte

D i s s e r t a t i o n

von Stefan Alexander Riemer
aus Berlin

1. Referent: Professor Dr. Dietmar Schomburg

2. Referent: Professor Dr. Dieter Jahn

eingereicht am: 24.04.2013

mündliche Prüfung (Disputation) am: 24.06.2013

Druckjahr 2013

Vorveröffentlichungen der Dissertation

Teilergebnisse aus dieser Arbeit wurden mit Genehmigung der Fakultät für Lebenswissenschaften, vertreten durch den Mentor der Arbeit, in folgenden Beiträgen vorab veröffentlicht:

Publikationen

Riemer SA, Rex R, Schomburg D: **A metabolite-centric view on flux distributions in genome-scale metabolic models.** *BMC Syst Biol* 2013, **7**:33.

Posterbeiträge

Riemer A, Schomburg, D: **metano – A Toolkit for Metabolic Network Analysis and Optimization.** (Poster) *German Conference on Bioinformatics* 2010, Braunschweig, Germany.

Gewidmet meinem Großvater Dr. Bodo Reblin.

Table of contents

Zusammenfassung

Metabolische Modelle sind wichtige systembiologische Werkzeuge, um den Stoffwechsel von Zellen zu verstehen. Sie werden unter anderem zur Vorhersage von Angriffspunkten im Stoffwechsel von pathogenen Mikroorganismen sowie zur Optimierung von mikrobiellen Produktionsstämmen eingesetzt. Mit der Flux-Balance-Analyse und verwandten Methoden stehen Algorithmen zur Verfügung, mit denen das Verhalten metabolischer Netzwerke, sowohl von Wildtyp-Organismen als auch von Mutanten, unter verschiedenen Umweltbedingungen vorhergesagt werden kann. Existierende Implementierungen dieser mathematischen Methoden weisen allerdings verschiedene Nachteile auf, u.a. fehlende oder schwierige Erweiterbarkeit und die Zuweisung von unphysiologisch hohen zyklischen Flüssen in der Flux-Balance-Analyse. Letztere wird meist durch sekundäre Optimierungsstrategien unter zusätzlichem Rechenaufwand behoben. Daneben werden Flussverteilungen in metabolischen Netzwerken bisher überwiegend unter einer rein reaktionszentrischen Sicht analysiert, obwohl einzelne Reaktionsflüsse für sich kaum interpretierbar sind.

In der vorliegenden Arbeit wird *metano*, eine Sammlung von mathematischen Werkzeugen für die Analyse und Verfeinerung von metabolischen Modellen, vorgestellt. *metano* enthält zunächst effiziente Implementierungen von etablierten Methoden (neben der Flux-Balance-Analyse u.a. Minimization of Metabolic Adjustment und Flux Variability Analysis). Die Flux-Balance-Analyse wurde dabei so implementiert, dass das Problem biologisch unplausibler zyklischer Flüsse vollständig vermieden wird. Durch die Bereitstellung als Open Source und die objektorientierte Implementierung kann *metano* leicht erweitert und an geänderte Anforderungen angepasst werden. Innerhalb dieser Plattform wurde eine Reihe von neuen Algorithmen entwickelt, darunter die automatische Plausibilitätsprüfung von metabolischen Modellen sowie Methoden für die metabolitzentrische Analyse von Flussverteilungen. Durch die automatische Plausibilitätsprüfung wird es möglich, im Prozess der Erstellung und Verfeinerung von metabolischen Modellen Fälle zu erkennen, in denen Modifikationen am Modell zu qualitativen Änderungen im metabolischen Verhalten führen.

Die in dieser Arbeit entwickelten Methoden Split-Ratio-Analyse und Metabolitfluss-Minimierung etablieren eine metabolitzentrische Sicht auf Flussverteilungen, welche die klassische reaktionszentrische Sicht komplementiert. Durch die Analyse von Reaktionsflüssen in einem natürlichen Kontext erlaubt die metabolitzentrische Sicht eine tiefergehende biologische Interpretation als die alleinige Betrachtung der Reaktionsflüsse. So erlauben diese Algorithmen die Analyse aller Flüsse in einem Metabolitpool oder einem metabolischen Zyklus auf einen Blick, und der zelluläre Energiestoffwechsel kann durch Analyse der Aufspaltungsverhältnisse der Flüsse in weniger als zehn Knotenpunkten des Netzwerks charakterisiert werden. Die neu entwickelten Methoden wurden auf publizierte genomweite metabolische Modelle der Bakterien *Escherichia coli* und *Thermotoga maritima* angewendet. Durch die metabolitzentrische Analyse konnten mehrere Ansatzpunkte zur Optimierung der Modelle hinsichtlich ihrer Vorhersagen für spezifische Wachstumsbedingungen identifiziert werden.

Abstract

Metabolic models are important systems biological tools for understanding cellular metabolism. They are used, inter alia, for the prediction of drug targets in the metabolic networks of pathogenic microorganisms and for the optimization of microbial production strains. Flux balance analysis and related mathematical methods allow predicting the steady-state behavior of the metabolic networks of both wild-type organisms and mutants under different environmental conditions. Existing implementations of these algorithms suffer from several drawbacks, among others the lack or difficulty of extensibility and the assignment of unphysiologically high cyclic fluxes in flux balance analysis. The latter is usually corrected via secondary optimization strategies at the cost of increased computation time. To date, flux distributions in metabolic networks are analyzed predominantly under a reaction-centric framework, even though individual reaction fluxes do not have a clear interpretation by themselves.

In this thesis, *metano*, a collection of mathematical tools for the analysis and refinement of metabolic models, is introduced. *metano* contains efficient implementations of established computational methods (most importantly flux balance analysis, minimization of metabolic adjustment, and flux variability analysis). Flux balance analysis has been implemented in such a way that the problem of biologically implausible cyclic fluxes is avoided completely. As it is distributed as open source and has been implemented using the object-oriented approach of software engineering, *metano* can easily be extended and adapted to changed requirements. Within this software framework, several new algorithms have been developed, including the automated plausibility checking of metabolic models and methods for the metabolite-centric analysis of flux distributions. In the process of model construction and refinement, the automated plausibility checking enables the identification of cases where a modification to the model leads to qualitative changes in the metabolic behavior.

The methods of split-ratio analysis and metabolite flux minimization, which were developed in this work, establish a metabolite-centric view on flux distributions. This view complements the classical reaction-centric framework. By analyzing reaction fluxes in a natural context, the metabolite-centric view allows a more profound biological interpretation than is possible from the consideration of isolated reaction fluxes. These algorithms allow the analysis of all fluxes in a metabolite pool or a metabolic cycle at a glance, and the energy metabolism of the cell can be characterized by considering the split ratios in less than ten branch points of the network. The newly developed methods were applied to published genome-scale metabolic models of the bacteria *Escherichia coli* and *Thermotoga maritima*. By performing a metabolite-centric analysis, it was possible to identify several prospects for refinement of the models with respect to their predictions for specific growth scenarios.

1 Introduction

For more than two decades, genome-scale metabolic models have been used to predict the metabolic capabilities of organisms, both wild type and mutants, in different environments. The genome-scale reconstruction of metabolic networks has been enabled by advances in genome sequencing techniques, the collection and sharing of *omics* data in electronic databases, and the development of computational tools for the analysis of model structure and behavior. Genome-scale metabolic models have been used, inter alia, for the identification of novel drug targets in pathogenic micro-organisms [1–4] and cancer cells [5], for predicting adverse side effects of drugs [6], and for optimizing microbial strains for the biotechnological production of chemicals of industrial or medical importance [7–10]. Genome-scale models have further been used to study robustness [11] and evolution [12] in metabolic networks.

Flux balance analysis (FBA) is one of the most widely used computational methods for the analysis of metabolic models [13,14]. It predicts the metabolic fluxes in a stoichiometric network model of the metabolism of a cell under certain biologically motivated assumptions using linear programming [15–17]. In the last decade, a large number of additional algorithms have been developed for the analysis of genome-scale metabolic models and the application of such models to different biological and biotechnological problems [18,19]. At the same time, a multitude of software programs implementing FBA and related computational methods have been released (see [19] for an overview).

Existing software solutions for the analysis of genome-scale metabolic models have several drawbacks: While many programs are free for academic users, most are still proprietary. Truly open-source software remains relatively rare in this field. In addition, many implementations that are open source still depend on commercial software such as MATLAB[1], CPLEX[2], LINDO[3], or MOSEK[4] for solving mathematical optimization problems.

Furthermore, many existing software tools are web-based. Web-based applications require the user to upload their (unpublished) models and data to the server of the software's operator. This is clearly unacceptable in commercial research and development. Moreover, it is uncommon for academic research groups that are not formally collaborating to exchange substantial amounts of unpublished data, much less entire models.

1 The MathWorks Inc., http://www.mathworks.com/products/matlab/

2 IBM Corporation, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

3 LINDO Systems Inc., http://www.lindo.com

4 MOSEK ApS, http://mosek.com/products/mosek/

Closed-source software and web-based applications limit their users to those applications envisioned by the developers, as the software's functionality cannot be extended. For scientific software, which is used for extending the boundaries of current knowledge, this is a severe, potentially crippling limitation.

This drawback is shared to some extent with open-source software that has not been designed from the start with the goal of extensibility. For instance, the software tool FASIMU [20] is written in the parser language AWK[1] and the shell script language bash[2]. Both of these programming languages are highly specialized – AWK for processing text files, and bash for automatizing operating system tasks. Shell script languages have a reputation for being 'write-only languages', and shell scripts are notoriously hard to maintain. In addition, knowledge of these specialist languages is not common among life scientists.

Finally, virtually all existing implementations of flux balance analysis assign extremely large, thermodynamically infeasible values to fluxes in stoichiometrically balanced reaction cycles [21]. Some implementations report these values in the computed flux distribution, leaving recognition and treatment of this problem to the user. Most existing programs, however, offer a secondary optimization procedure, which finds or eliminates the problematic cyclic fluxes automatically at the cost of increased computation time [22].

Metabolic models and predicted flux distributions are predominantly analyzed in a reaction-centric fashion [23]: To assess a model's quality, predicted reaction fluxes are compared to experimentally determined flux values, which are given either as absolute values in mmol gDW^{-1} h^{-1} or as percentages of the substrate uptake rate [24,25]. Gene knockouts are represented in the model by constraining the fluxes of all reactions catalyzed by the gene product(s) to zero [26]. In the same way, gene expression data can be incorporated by introducing or modifying constraints on reaction fluxes [27–30]. Finally, the most commonly employed computational strategies in metabolic engineering attempt to find a combination of gene knockouts and/or modify individual reaction fluxes to boost production (and secretion) of a desired chemical compound [31–34].

Reaction fluxes are, however, hard to interpret in isolation. To give an example, no insights can be gained from the fact that in *Escherichia coli*, a glutamate-producing flux of 6.3 mmol gDW^{-1} h^{-1} through glutamate dehydrogenase has been predicted for a certain growth scenario. Expressing this flux as a fraction of the glucose uptake rate (57%) does not overcome this difficulty. A promising approach to the interpretation of the flux through a given reaction may be to express the flux as a ratio of the total flux through each of the reactants and products involved in the reaction. For glutamate dehydrogenase, e.g., the relevant questions are what fraction of the L-glutamate is produced and what fraction of the NADPH is consumed via this particular reaction.

1 GNU awk, The GNU Project, http://www.gnu.org/software/gawk

2 GNU Bash, The GNU Project, http://www.gnu.org/software/bash

With flux-sum analysis a first metabolite-centric approach for the analysis of genome-scale metabolic models has been proposed [11,35]. This approach has, however, remained limited to the analysis of network robustness [11,36]. The examination of metabolic branch points and flux split ratios is very important in the field of metabolic engineering [37–39]. Branch points and split ratios have been considered even longer in wet-lab metabolic flux analysis [40–42]. However, all existing approaches use only small metabolic networks and look only at a small number of predefined branch points. The systematic analysis of genome-scale metabolic networks requires a non-targeted automated analysis of branch points and split ratios.

The goals of this work were threefold: Firstly, to develop an easily extensible open-source toolkit implementing the most commonly used mathematical methods for the analysis of genome-scale metabolic models. This software toolbox is named *metano*. Secondly, to develop an implementation of flux balance analysis that avoids the problem of thermodynamically forbidden cycles entirely. And thirdly, to develop and implement within the new framework a set of new computational methods for the analysis and refinement of metabolic models. The focus in the development of new algorithms was placed on two areas in particular, the high-quality reconstruction of metabolic networks and the metabolite-centric analysis of flux distributions.

This thesis is structured as follows: Chapter 2 introduces basic concepts and definitions as well as the theoretical background of metabolic modeling. Metabolic modeling is presented within the context of the wider field of systems biology. In addition, this chapter expounds on the problem of thermodynamically infeasible loops outlined above. The model organisms considered in this work are also introduced in Chapter 2. Chapter 3 presents the methods implemented in *metano* and the published metabolic models to which these methods are applied in the subsequent chapters. In Chapters 4 and 5, the main results of this work are presented and discussed. Chapter 4 describes *metano* and gives details on the implementation. In addition, the validation of the implementations of established methods in *metano* is presented in Chapter 4. In Chapter 5, the application of *metano* to the analysis of two published metabolic models is presented. Results are described and discussed in the same chapter. Chapter 6 describes, in more general terms, the applications of *metano* in the reconstruction of metabolic networks. Chapter 7 gives a summary and conclusion.

2 Background

2.1 Systems biology

Life is a hugely complex phenomenon that entails a plethora of interlocking processes acting on vastly different size and time scales – from ångströms to meters and from picoseconds to days. Traditionally, researchers have examined biological processes under a reductionist view, trying to gain as much knowledge as possible about isolated components of biological systems. Both the fundamental building blocks of the cell and the basic functional modules of the molecular machinery of the cell have been studied extensively in this fashion. Cellular building blocks include, among others, DNA, RNA, proteins, lipids, polysaccharides, and small molecules. The studied elementary cellular processes include DNA replication, transcription, translation, as well as individual metabolic and signaling pathways.

In the last decades, this reductionist view has gradually been replaced by a more systems-oriented approach, which has been termed systems biology [43]. This paradigm shift has been necessitated by the realization that the components of living systems are less separable than initially thought. This is exemplified by inherently systemic phenomena such as crosstalk between signal transduction pathways [44], the highly complex protein-protein interaction networks at work in biological systems [45], and the large number of ways in which transcription of a gene can be regulated, many of which can be active at the same time [46,47]. The systems-oriented study of living systems has been made possible by the advent of high-throughput techniques in the related *–omics* disciplines [48]. These fields include genomics, transcriptomics, proteomics, metabolomics, and many more. The focus of systems biology is the integration of *omics* data obtained by different experimental techniques with the goal of obtaining a better understanding of biological systems.

Systems biologists use mathematical modeling to generate a functional description of the system under consideration [49]. Computational models can be used to simulate processes that underlie measurable phenomena but cannot be observed directly. Models are valuable tools for generating hypotheses, which can then be used to guide wet-lab experiments [50]. Experimental results can in turn be used to refine the model. The resulting cycle of data- and hypothesis-driven modeling and hypothesis-driven design and execution of experiments is shown schematically in Figure 1. It should be noted that the early stages of model refinement do not usually entail wet-lab experiments. If the model is still internally inconsistent or if its predictions are in disagreement with published experimental data, these conflicts have to be resolved first before the model's predictions can be used to guide new experiments. This approach is highly economic, as wet-lab experiments are usually much more costly in time, space, energy, and material requirements than computer simulations.

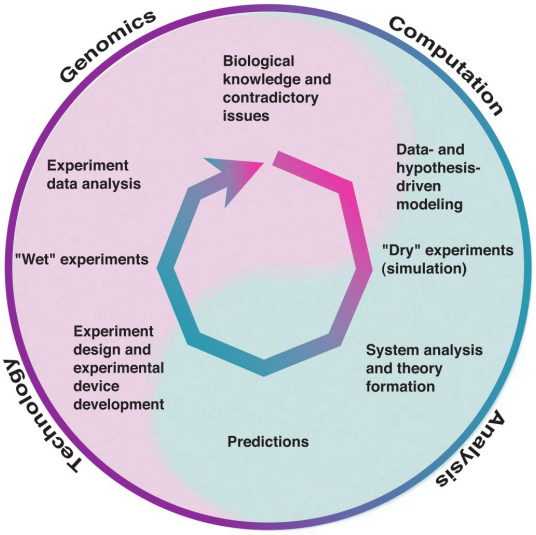

Figure 1: Cycle of hypothesis-driven research in systems biology. Figure taken from [49].

The analysis of mathematical models requires computational methods. The development of a wealth of new computational tools and the continuous increase in computing power have strongly contributed to the success of systems biology [51]. Recent years have seen the development of computational methods supporting or partially automatizing the process of model generation (e.g. [52–54]).

2.2 Metabolism as determinant of phenotype

The biochemical processes inside the cell can be described by a hierarchical scheme: Based on the static genome, the cell's transcription and translation machinery produces RNAs and proteins [55]. The set of all RNAs that are present in the cell at a given time is called the transcriptome, and the set of all proteins is called the proteome. A large number of proteins are enzymes, which catalyze the conversion of substrates to products in chemical reactions. The set of all these chemical reactions forms the metabolism of the cell, and the set of all metabolites present at a given time is called the metabolome. Figure 2 shows this hierarchical organization schematically along with the most important types of interactions between the three dynamic layers.

The three layers differ in their response time to perturbations: Changes in gene expression take between hours and days to produce an effect on the phenotype [56,57], protein-protein interactions act on an intermediate scale of seconds to minutes [58], while changes on the metabolic level alone can alter the phenotype very rapidly [59]. As nearly all changes in gene expression or protein modulation ultimately act by altering the cell's metabolism, metabolism is the most immediate determinant of the cellular phenotype [60].

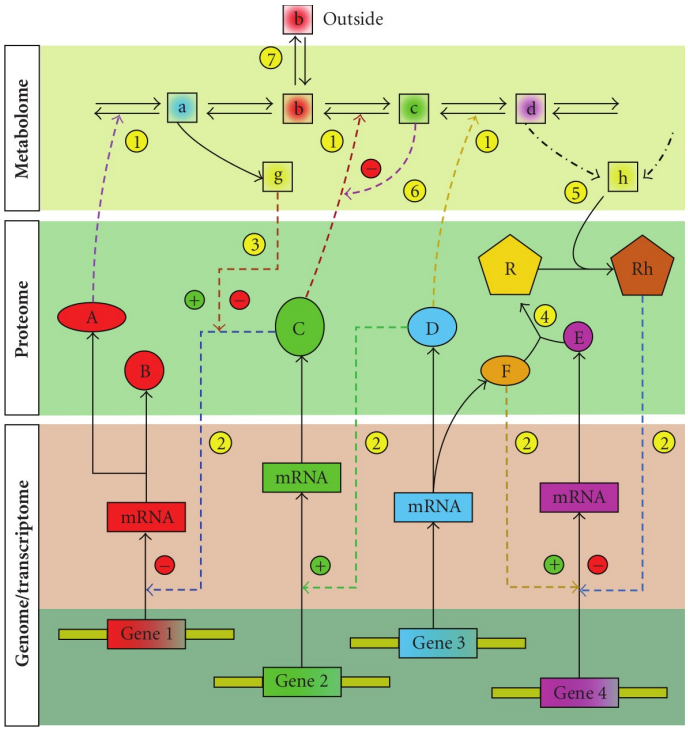

Figure 2: The biochemical interaction network of the cell has a hierarchical organization, which comprises the static genome and the three dynamic, mutually interdependent layers transcriptome, proteome, and metabolome. The following types of interactions are shown: (1) enzyme catalysis, (2) post-transcriptional control of gene expression, (3) effect of a metabolite on gene transcription mediated by a protein, (4) protein-protein interaction, (5) effect of a downstream metabolite on transcription via binding to a protein, (6) feedback inhibition or activation of an enzyme by a downstream metabolite, and (7) transport of a metabolite into or out of the cell. Figure taken from [48].

2.3 Metabolic modeling

A metabolic model is a network of chemical reactions transforming sets of chemical compounds into one another. In this thesis, all chemical compounds involved in metabolic reactions are referred to as metabolites, regardless of size or subcellular localization. Metabolic networks have been modeled on three levels of detail: graph-based, stoichiometric, and kinetic.

Firstly, a metabolic network can be represented as a bipartite graph, where reactions and metabolites are modeled as distinct types of nodes. A reaction is described in the graph by a reaction node and the edges connecting it to the metabolite nodes corresponding to the compounds involved in the reaction. Panels A and B of Figure 3 show an example network comprising four reactions and three metabolites as chemical reaction equations (A) and as a bipartite graph (B). Graph-based approaches focus on the study of topological properties of metabolic networks [61,62]. Within a purely graph-theoretic framework, reaction stoichiometry cannot be represented, and graph-based approaches are not suitable for studying dynamic system behavior [63]. Graph-theoretic algorithms have, among other applications, been used to study redundancy and robustness [64], to

decompose networks into functional modules [65], and to identify critical points in metabolic networks [66].

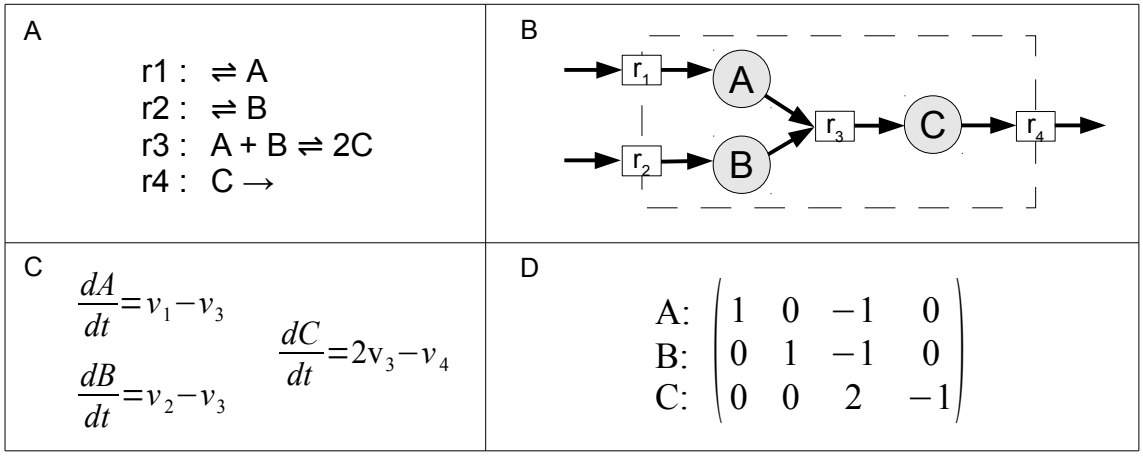

Figure 3: Four representations of a small example metabolic network: (A) Chemical reaction equations. (B) Bipartite graph (dashed lines represent the system boundary). (C) System of ordinary differential equations describing the time derivatives of the concentrations as functions of the reaction fluxes (v_1 is the flux through reaction r_1, v_2 through reaction r_2, etc.). (D) Stoichiometric matrix.

Secondly, metabolic networks can be modeled in kinetic detail. Kinetic models represent the network as a system of ordinary differential equations [67]. An example of this is shown in Figure 3C: The time derivatives of the metabolite concentrations are linear functions of the fluxes v_1 through v_4 corresponding to the reactions r_1 through r_4 shown in Figure 3A. Kinetic models can, in principle, predict the system's dynamic response to perturbations and the time-dependent behavior of other non-steady-state processes [68]. Their greatest limitation lies in the availability of kinetic parameters, which are required for all reactions in the network. In general, kinetic parameters are available for only a small number of enzymes in the network, and *in-vivo* reaction rates may be substantially different from those observed *in vitro* for isolated enzymes [69,70]. Hence, the construction of fully kinetic metabolic models on a genome-wide scale seems infeasible at present [71]. Despite their limitations, kinetic models are valuable tools for studying non-steady-state processes, most importantly signal transduction [72,73] and the cellular response to environmental stress [74]. Moreover, approaches for combining stoichiometric and kinetic models have been proposed towards the goal of building an integrated whole-cell model [75].

In the third commonly used approach to metabolic modeling, the network is represented by a set of reaction equations, such as that shown in Figure 3A. Stoichiometric models are characterized by the condition of mass balance, where in each reaction, each kind of atom appears in the same number on the left-hand side as on the right-hand side. The only exception to this condition are reactions transporting matter over the system boundary into or out of the system under consideration. As stoichiometric models incorporate reaction stoichiometry but do not require knowledge of kinetic parameters, this level of detail is well suited for the reconstruction of whole-genome metabolic networks [7]. In fact, all genome-scale metabolic models published to date are stoichiometric models [18]. Stoichiometric models do not allow the analysis of

non-equilibrium dynamics, but have proven very useful for predicting the metabolic capabilities in steady state [76–79]. The remainder of this thesis will focus exclusively on stoichiometric metabolic models.

Kinetic and stoichiometric metabolic models allow the prediction of fluxes under certain assumptions. This allows the identification of essential reactions (and thus essential genes) and the characterization of the split ratios of metabolic fluxes in the network. By examining the variability of reaction fluxes, it is possible to identify alternative metabolic pathways. As both are deterministic models, a kinetic or stoichiometric model does not represent an individual cell but rather an 'average' cell in a larger population [80].

2.3.1 Stoichiometric matrix and steady state

A stoichiometric model with n reactions and m metabolites is represented mathematically by an $m \times n$ stoichiometric matrix **S**. Each row of **S** corresponds to one metabolite, and each column corresponds to one reaction. The entry s_{ij} of **S** is the stoichiometric coefficient of metabolite i in reaction j. It is positive if the metabolite occurs on the right-hand side of reaction j, negative if it occurs on the left-hand side, and zero if it is not involved in the reaction. Figure 3D on page 8 shows the stoichiometric matrix corresponding to the network defined by the reactions shown in Figure 3A. The stoichiometric matrix transforms the $n \times 1$ vector of reaction fluxes **v** into a vector containing the time derivatives of the metabolite concentrations [81]:

$$S \cdot v = \frac{dc}{dt}. \tag{1}$$

See Figure 3C for an example of such a system of ordinary differential equations. The flux v_j through reaction j is positive if the reaction proceeds from left to right and negative if the reaction proceeds in the reverse direction.

The metabolic steady state is a flux equilibrium, in which all metabolite concentrations are constant [82]. This means that for each metabolite node in the system, the sum of all incoming (i.e. producing) fluxes is equal to the negative sum of the outgoing (i.e. consuming) fluxes. In other words, all metabolites produced by fluxes in the network have to be consumed at the same time by other fluxes. The steady state is thought to exist for every metabolic network under constant external conditions, and most biochemical systems in nature are known to operate close to a steady state [83].

All analyses of stoichiometric metabolic networks inherently assume that the system under consideration is in steady state. The steady-state condition is expressed mathematically as

$$S \cdot v = 0. \tag{2}$$

In steady state, the system of ordinary differential equations defined by Equation 1 simplifies to a system of linear equations (Equation 2). As metabolic networks contain more reactions than metabolites, this system is underdetermined [84]. This means that the solution space of Equation 2 (i.e. the null space of **S**) is infinitely large. Computational methods for the analysis of metabolic models either try to explore the shape of the solution space or to find a particular solution that is most biologically meaningful under certain conditions.

2.3.2 Genome-scale metabolic models

With the availability of rapid sequencing technologies, electronic databases for the collection and sharing of *omics* data, and suitable computational tools, it has become possible to develop whole-genome reconstructions of metabolic networks.

Firstly, a genome-scale metabolic model is a structured knowledge base that integrates data about the metabolism of the organism under consideration from different sources, including databases, primary literature, and expert knowledge [85]. Secondly, genome-scale models allow the prediction of cellular phenotypes. Among other things, this allows the *in-silico* prediction of the effects of genetic or regulatory modifications or growth in different environments on the phenotype [23]. The predicted metabolic capabilities of a genome-scale network in a certain scenario can be used for hypothesis-driven planning and execution of wet-lab experiments and for providing context for the interpretation of experimental data.

Important applications of genome-scale metabolic models include the prediction of potential drug targets in pathogenic microorganisms [2] and the study of environmentally triggered phenotypic changes. Examples of such transitions that have been studied using metabolic models are a pathogen or parasite switching from an insect vector to a mammalian host [86] and a non-pathogenic soil bacterium becoming pathogenic inside a mammalian host [87]. Studying the metabolism of pathogenic organisms in comparison with host cell metabolism can lead to the identification of novel drug targets [88]. Furthermore, an accurate model of cellular metabolism allows metabolic engineering, the directed modification of the organism with respect to a metabolic goal, such as overproduction and export of a medically or commercially relevant chemical compound [89].

2.3.3 System boundary and compartments

In a genome-scale metabolic model, the system under consideration usually includes the part of the extracellular space immediately surrounding the cell. Hence, the model contains at least two compartments: intracellular and extracellular. In general, there is one compartment in the model for each membrane-enclosed compartment in the cell. In addition, virtual compartments may be used to represent other reaction spaces with limited accessibility, such as for membrane-bound reactions, or reactions with unknown

subcellular localization [90]. Figure 4 schematically shows a system with three compartments.

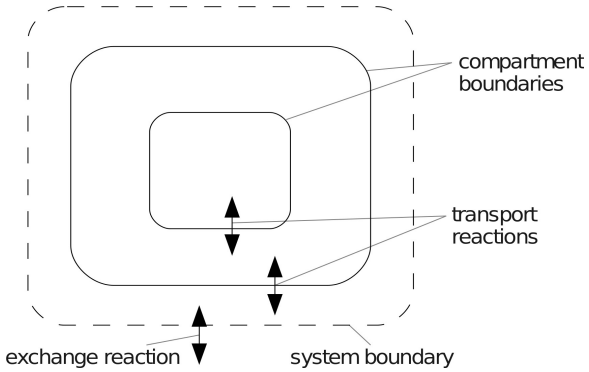

Figure 4: A system with two nested cellular compartments and one extracellular compartment.

Compartments are modeled by duplication of metabolite nodes, which means that pools of the same metabolite occurring in different compartments are represented by separate entities (metabolite nodes). For instance, a model of a Gram-negative bacterium might contain metabolite nodes labeled 'glucose[extracellular]', 'glucose[periplasm]', and 'glucose[cytosol]' for representing pools of D-glucose in the respective compartments. The transport of metabolites between compartments is modeled by virtual transport reactions, which convert between a metabolite in one compartment and the same metabolite in another compartment. Table 1 shows examples of these virtual reactions.

Table 1: Examples of virtual reactions representing the transport of metabolites between compartments. (Unmarked metabolites are localized in the cytosol.)

Type of transport	Example
Passive diffusion	O_2[extracellular] ⇌ O_2
Passive uniporter	glucose[extracellular] ⇌ glucose
Symporter (active or passive)	acetate[periplasm] + H^+[periplasm] ⇌ acetate + H^+
Antiporter (active or passive)	succinate + citrate[periplasm] ⇌ citrate + succinate[periplasm]
Primary active transport (ABC transporter)	L-alanine[extracellular] + ATP + H_2O → L-alanine + ADP + phosphate + H^+
Group translocation (phosphotransferase system)	glucose[extracellular] + phosphoenolpyruvate → glucose 6-phosphate + pyruvate

In steady state, the system is regarded as surrounded by an infinite environment with constant composition. This condition corresponds best to exponential growth in a bioreactor in continuous mode, where fresh medium is supplied at a constant rate. The exchange of metabolites with a larger environment is represented in the model by a set of exchange reactions, which transport metabolites from the environment to the modeled part of the extracellular space and vice versa. The exchange reactions are not stoichiometrically balanced: They contain the transported metabolite on one side, while

the other side is empty. In the toy example shown in Figure 3 on page 8 (which has only a single compartment), reactions r_1, r_2, and r_4 are exchange reactions, while reaction r_3 is an internal (i.e. metabolic) reaction.

2.3.4 Constraint-based modeling

The construction of stoichiometric metabolic models is an instance of constraint-based modeling, where the solution space of a mathematical problem is defined via a set of equality and inequality constraints [91]. This is in contrast to theory-based modeling, which has the goal of determining a single 'exact' solution [81]. For stoichiometric models, the relevant equality constraints are given by the steady-state condition (Equation 2 on page 9), while the most common type of inequality constraints has the form

$$\mathrm{lb} \leq v \leq \mathrm{ub}\,. \tag{3}$$

This means that the reaction fluxes **v** are constrained via two $n \times 1$ vectors **lb** and **ub** of lower and upper bounds.

Inequality constraints are used to model the following, among others:

- Irreversibility of reactions (lower or upper bound set to zero),
- measured flux values (e.g. bounds = mean $\mp$ standard error),
- known uptake or secretion rates, and
- known split ratios of fluxes.

The solution space defined by the equality and inequality constraints is the set of all feasible metabolic states. It does not contain an intrinsic measure of the likelihood of any particular flux distribution in a given environmental scenario [92]. Therefore, any computational method for predicting the metabolic capabilities of a stoichiometric model needs to make additional assumptions to further constrain the solution space in order to produce biologically meaningful predictions.

2.3.5 Biomass reaction, units, GAM, and NGAM

The growth of the cells is modeled as flux through a virtual biomass reaction. This flux will hereafter be referred to as biomass flux for short. The biomass reaction contains on the left-hand side all metabolites that are known biomass components of the modeled organism in correct molar ratios. In the case of polymers, monomer precursors are given instead. The molar ratios of the biomass components are given as absolute values in millimoles per gram dry weight. The use of absolute ratios leads to a normalization of the biomass such that one millimole of the virtual biomass metabolite produced by the biomass reaction corresponds to one gram dry weight. While all other reaction fluxes are in mmol $\mathrm{gDW^{-1}\,h^{-1}}$, the unit of the biomass flux is $\mathrm{h^{-1}}$ because of the normalization.

In steady state, the produced biomass leaves the system under consideration, which is often modeled in practice by omitting the virtual biomass metabolite on the right-hand side of the biomass reaction.

The energy requirement for cellular growth and reproduction also has to be considered in the model. It is included in the form of an ATP hydrolysis reaction whose flux is coupled to the biomass flux. The amount of ATP in millimoles that is required for the formation of one gram of dry biomass is called *growth-associated maintenance requirement* (GAM) [93]. Most of the GAM is accounted for by the synthesis of macromolecules such as DNA, RNA, and proteins. Usually, the GAM is included in the biomass reaction. This is achieved by adding z ATP + z H_2O to the left-hand side and z ADP + z phosphate + z H^+ to the right-hand side.

Certain cellular processes consume energy but are not associated with cell growth. These include, among others, DNA repair and maintaining osmotic pressure and membrane potential. They are modeled as a combined ATP hydrolysis reaction with a fixed flux. The flux through this reaction is called *non-growth-associated maintenance requirement* (NGAM) [94].

2.4 Thermodynamically infeasible loops

An infeasible loop, also called an internal cycle, a stoichiometrically balanced cycle, or a type III extreme pathway, is a set of reactions that is capable of transforming a group of metabolites into itself via a series of conversions without any loss or gain of matter [95,96]. In the example shown in Figure 5, the sequence of reactions 2–3–4–1 is capable of transforming metabolite C back into itself.

R1 : A + B ⇌ C

R2 : C + ATP ⇌ CP + ADP

R3 : CP + ADP ⇌ B + D + ATP + H

R4 : D + H ⇌ A

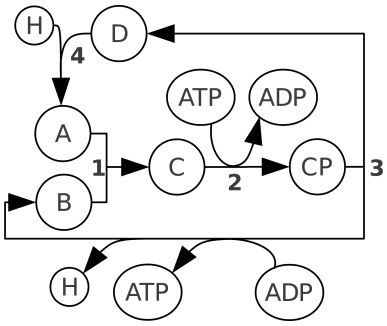

Figure 5: A thermodynamically infeasible loop (left: reactions, right: network). Metabolite C can be converted back to itself in four reaction steps.

In steady state, biochemical reactions can only proceed in the direction of decreasing free energy [95]. As the net change in free energy associated with any stoichiometrically balanced cycle of chemical reactions is zero, cyclic fluxes are thermodynamically forbidden. If any reaction in the network is known to proceed in the direction converting a set of substrates S to a set of products P in the scenario under study, then it follows from Hess's law [97] that there can be no net flux through a sequence of reactions

transforming P to S without consumption of energy. If, in the example of Figure 5, reaction 2 is known to proceed in the direction producing CP and ADP, at least one of the reactions 1, 3, and 4 must have a non-positive flux (i.e. either R1 : A + B ← C, R3 : CP + ADP ← B + D + ATP + H, or R4 : D + H ← A).

As the reactions forming an infeasible loop are stoichiometrically balanced, cyclic flux through the loop does not result in any net production or consumption of metabolites. Therefore, adding any amount of such cyclic flux to a steady-state flux distribution preserves the steady-state condition. Restricting the solution space further by adding inequality constraints can only partially eliminate this ambiguity. For instance, in the example above, reaction 2 might proceed in one direction in some environmental conditions and in the opposite direction in others. Furthermore, it might not be known which other reaction in the cycle must necessarily proceed in the opposite direction.

In particular, flux distributions in metabolic networks predicted via mathematical optimization strategies frequently contain arbitrarily large cyclic fluxes through infeasible loops, as the objective functions commonly employed are blind to such cyclic fluxes. A trivial workaround would be to secondarily minimize the total flux through the network [98] or the total number of reaction steps [99]. In addition, a large body of work has been published in the last decade on identifying and eliminating infeasible loops in metabolic models (e.g. [22,95,96,100–103]).

2.5 Comparability of model predictions and experimental observations

2.5.1 Experimental data that can guide model validation and refinement

Several types of high-throughput *omics* data can be used in the process of model construction, validation, and refinement. These include genome, transcriptome, proteome, protein-protein interaction, metabolome, fluxome, and phenotypic data [23].

The process of metabolic modeling always starts from a genome annotation [104,105]. All enzyme functions that have been described for the studied organism have to be included in the model. Usually, all enzymes identified in the genome with a high confidence score are added in an automatic step. Commonly used annotation sources include biochemical databases such as BRENDA [106], KEGG [107], or UniProtKB [108]; pattern-based similarity searches [109]; and BLAST-based homology searches [105].

The information obtained from genome annotation data can and should be refined by incorporating gene expression data. This ensures that only genes (and the reactions catalyzed by their products) are included that are actually expressed in the scenario under consideration. When comparing metabolic behavior in different growth scenarios or between wild-type and perturbed networks, differential gene expression data can further be used to formulate scenario-specific constraints. For instance, if the expression of an enzyme is reduced by 50% in the studied scenario compared to the reference condition and the reaction catalyzed by that enzyme is known to be the rate-determining

step in a biosynthetic pathway, an upper bound of 50% of the value for the reference scenario can be set for the flux through the reaction. Data on gene regulation or regulatory protein-protein interactions can be incorporated in a similar fashion.

Metabolome data can be employed for model refinement. In an untargeted metabolomic analysis, it is routinely possible to detect several hundred metabolites, of which about 100 to 150 can be identified [110,111]. First of all, for each metabolite detected in a specific growth scenario, the model must predict at least one reaction involving the metabolite to be active. Furthermore, absolute metabolite concentrations can be used in conjunction with Gibbs energies of formation to predict the directions in which reversible reactions operate in the scenario under study [112,113].

The primary variables in a metabolic model are reaction fluxes, which are directly comparable to fluxome data. The most common method of determining fluxes *in vivo* is tracer analysis, where the organism is grown on stable-isotope-labeled substrates for a defined period of time, after which the distribution of the label in the metabolome is analyzed. The label distribution is determined using either mass spectrometry, usually after a chromatographic separation step, or nuclear magnetic resonance spectroscopy [114]. The isotopes employed in tracer experiments include ^{2}H, ^{13}C, ^{15}N, ^{18}O, and ^{31}P [114,115], with ^{13}C studies being by far the most common [116–118]. Traditional approaches to fluxomics are based on targeted metabolomics, studying only the labeling patterns of a defined set of metabolites, such as amino acids [119]. A more recent strategy employs untargeted metabolomics and evaluates the labeling patterns of all detectable compounds [120]. As all fluxomics approaches depend on metabolomics, only those reactions can be studied that involve metabolites with sufficiently high concentrations. Therefore, ^{13}C tracer experiments can, in general, only determine fluxes in or close to the central carbon metabolism [117,118,121].

Finally, phenotypic data can be used to qualitatively validate model predictions for different growth scenarios. Among the most important experimental tools for obtaining high-throughput phenotypic data are phenotype microarrays [122]: Non-selectively precultured cells are transferred to a microtiter plate, each well of which contains a different cultivation medium and a redox dye. The originally colorless dye is reduced by enzymes of the electron transfer chain of the cell using NADH as electron donor [123], provided that the cell is capable of respiration in the medium. Thus, a color change is observed for each usable medium. By taking photographic measurements at regular intervals, kinetic curves can be obtained [123]. Phenotypic data are usually incorporated in the model validation process in the form of binary variables (growth/ no growth on a particular substrate or combination of substrates) [23]. It should be noted that phenotype microarrays do not measure growth but respiration, which are not necessarily proportional.

2.5.2 Quantitative prediction of cell growth

If the biomass reaction reflects the correct biomass composition and is normalized (as described in Section 2.3.5) and the values of substrate uptake rate, GAM, and NGAM

(see Section 2.3.5) have been fitted to experimentally determined values, the flux through the biomass reaction corresponds to the specific growth rate μ of exponentially growing cells. Exponential (unlimited) growth occurs, e.g., in a bioreactor in continuous mode or during the exponential phase of growth in a limited environment. The specific growth rate is defined as the increase in cell mass x per unit time per gram cell mass [124,125]:

$$\mu = \frac{dx}{dt} \cdot \frac{1}{x}. \tag{4}$$

As the unit of $\frac{dx}{dt}$ is [unit of x] per hour, the unit of x cancels out, and the unit of μ is h^{-1}. As cell mass is hard to determine experimentally (impossible *in vivo*), other measures are routinely used, which are proportional under certain assumptions. These are biomass concentration (g L^{-1}; for constant volume (such as in a bioreactor or shaker flask)), cell number (if cell size is constant), and optical density (if cell size, cell shape, and cell envelope composition are constant). The specific growth rate can be calculated from an experimentally determined growth curve by measuring cell mass x (or, more commonly, a proportional measure such as optical density) at two points in time t_1 and t_2 during a period of exponential growth:

$$\mu = \frac{\ln x_2 - \ln x_1}{t_2 - t_1}, \tag{5}$$

where x_1 and x_2 denote cell mass at time t_1 and t_2, respectively [124].

The equivalence of biomass flux and specific growth rate during exponential growth is obtained from the following consideration:

The stoichiometric matrix **S** of the model transforms a flux vector **v** into the corresponding vector of the time derivatives of the metabolite concentrations (Equation 1 in Section 2.3.1). The components of this vector are normalized to one gram of dry biomass, like the reaction fluxes. They are not time derivatives of the absolute concentrations in mmol L^{-1}.

In particular, this also applies to the time derivative of the concentration of the virtual biomass metabolite (see Section 2.3.5), which is determined by the biomass flux as follows:

$$\frac{dc_{\mathrm{BM}}}{dt} = v_{\mathrm{BM}}. \tag{6}$$

This normalized derivative is obtained from the time derivative of the absolute biomass concentration (or cell mass) x by dividing the latter by the current value of x:

$$\frac{dc_{BM}}{dt} = \frac{dx}{dt} \cdot \frac{1}{x}. \quad (7)$$

The right-hand sides of Equations 4 and 7 are identical. Therefore, with Equation 6 follows:

$$\frac{dc_{BM}}{dt} = v_{BM} = \mu. \quad (8)$$

2.6 Escherichia coli

Escherichia coli is a Gram-negative bacterium in the family Enterobacteriaceae in the class Gammaproteobacteria. The bacterium is rod-shaped and facultatively anaerobic. *E. coli* occurs as a commensal organism in the gut flora of birds and mammals. In addition, several human-pathogenic strains of *E. coli* exist, and commensal strains can become pathogenic in immunocompromised patients [126]. *E. coli* has long been used as a model organism in the life sciences, and it is by far the best-studied prokaryote. Many metabolic enzymes and pathways were originally discovered in *E. coli* (e.g. [127–130]), and its metabolism has been modeled in increasing scopes and levels of detail since the mid-1980s [76,131,132]. *E. coli* can easily be cultured in defined high-glucose minimal media. During rapid aerobic growth on glucose, *E. coli* secretes acetate. This phenomenon has been termed overflow metabolism [133]. The overflow effect is associated with transcriptional regulation of genes in the TCA cycle and the electron transport chain [134].

MG1655 is a laboratory strain of *E. coli* derived from the wild-type strain K-12, which was isolated in 1922 from human feces. MG1655 was the first *E. coli* strain to be sequenced; the genome sequence was published in 1997 [135].

2.7 Thermotoga maritima

Thermotoga maritima is a rod-shaped hyperthermophilic bacterium in the order Thermotogales, which is considered one of the deepest branches within the Bacteria [136]. The cell membrane of *T. maritima* is a lipid monolayer, and the cells exhibit Gram-negative staining [137]. *T. maritima* is an obligately anaerobic heterotroph, obtaining the energy required for growth via fermentation of carbohydrates [138]. Its optimal growth temperature is 80°C [139], and it is a model organism for the study of hyperthermophilia and of phylogenetically ancient bacteria [140,141].

MSB8 (DSM 3109) is the type strain of *T. maritima*. It was isolated from geothermally heated marine sediment at Vulcano, Italy and first described in 1986 [142]. The genome sequence was published in 1999 [143].

2.8 Pseudomonas aeruginosa

Pseudomonas aeruginosa is an obligately respiring bacterium [144] in the family Pseudomonadaceae in the class Gammaproteobacteria, which can thrive in a wide range of environments. Cells are rod-shaped and stain Gram-negative. *P. aeruginosa* is an opportunistic human pathogen that can cause, among others, lung and urinary tract infections [145]. Of biotechnological interest is its capability to produce biofilms [146].

PAO1, a wound isolate first described in 1955 [147], is the most widely used laboratory strain of *P. aeruginosa*. Its genome sequence was published in 2000 [148].

3 Materials and methods

3.1 Genome-scale metabolic models

The computational methods and software implementations presented in this thesis were first validated against the whole-genome metabolic reconstruction iAF1260 of *Escherichia coli* [24]. The software was then used for detailed analyses of the published genome-scale metabolic models of two bacteria: *Escherichia coli* and *Thermotoga maritima*.

3.1.1 *Escherichia coli* K-12 MG1655

Two published genome-scale metabolic models of *E. coli* were used in this work: iAF1260, published in 2007 [24], and the more recent version designated iJO1366, which was published in 2011 [149]. As an updated version of the same model, iJO1366 was used for all actual *in-silico* analyses of metabolic phenotypes performed in this work. However, as the authors of iAF1260 provided flux distributions obtained using their model, this model was used for the validation of the implementation of established tools in *metano*.

3.1.1.1 iAF1260

The model iAF1260 covers 1,260 open reading frames of the genome of *Escherichia coli* K-12 MG1655. The model is an extension of the earlier model iJR904, which was published in 2003 by the same research group [77]. iAF1260 is based on the genome sequence published by Blattner et al. [135], a published functional annotation of *E. coli* [150], the EcoCyc database [151], and a global thermodynamic analysis of *E. coli* metabolism [152]. The model comprises three compartments (cytosol, periplasm, and extracellular space), 2,382 reactions (1,387 metabolic reactions, 690 transport reactions, 304 exchange reactions, and 1 biomass reaction), and 1,668 metabolite nodes (951 cytoplasmic, 418 periplasmic, and 299 extracellular). The growth-associated maintenance requirement (GAM) is 59.81 mmol ATP gDW^{-1}, while the non-growth associated maintenance requirement (NGAM) is 8.39 mmol ATP gDW^{-1} h^{-1}.

The model was downloaded from the supplementary material of the corresponding publication [24] in the SBML format [153]. The two downloaded SBML files both contain the complete model, but the files differ in the lower and upper bounds set for the reaction fluxes. The different sets of flux bounds represent two different growth scenarios. Both scenarios correspond to growth in defined media with glucose as the sole carbon source, ammonium as the sole nitrogen source, phosphate as the source of phosphorus, and sulfate as the sulfur source.

In **scenario 1** (SBML file Ec_iAF1260_flux1.xml), glucose uptake is limited to a maximum rate of 8 mmol gDW^{-1} h^{-1}, and the oxygen uptake rate is limited to 18.5 mmol gDW^{-1} h^{-1}.

In **scenario 2** (SBML file Ec_iAF1260_flux2.xml), 152 reactions known to be inactive during growth on glucose are deactivated, and the ratio between proton-translocating and non-proton-translocating NADH dehydrogenase is set to 1:1. The latter constraint is not present in the SBML file, but is described in the publication [24]. In this scenario, the glucose uptake rate is limited to 11.0 mmol gDW^{-1} h^{-1}, and the maximum oxygen uptake rate is set to 18.2 mmol gDW^{-1} h^{-1}.

3.1.1.2 iJO1366

The model iJO1366 is an updated version of iAF1260, published in 2011 [149]. It covers 1,366 open reading frames of the genome. The model encompasses 2,583 reactions (1,473 metabolic reactions, 778 transport reactions, 330 exchange reactions, and 2 biomass reactions) and 1,805 metabolite nodes (1,039 cytoplasmic, 442 periplasmic, and 324 extracellular). Both growth-associated and non-growth-associated maintenance requirements are considerably lower than the values used in the earlier model iAF1260: The GAM is 53.95 mmol ATP gDW^{-1}, and the NGAM is 3.15 mmol ATP gDW^{-1} h^{-1}. The model was downloaded as an SBML file from the supplementary material of the corresponding publication [149]. This file only contains universally valid lower and upper bounds on the reaction fluxes.

The following scenario, which will be referred to as the **original scenario**, was used to simulate aerobic growth in a bioreactor in continuous mode in a high-glucose minimal medium: The glucose uptake rate was limited to 11.0 mmol gDW^{-1} h^{-1}, and acetate secretion was added at a forced rate of 6.4 mmol gDW^{-1} h^{-1}. These values were reported for exponential growth in a bioreactor [154], the scenario that corresponds best to the infinite environment assumed implicitly in steady-state analyses of metabolic models. In addition, the constraints described for iAF1260 specifically for aerobic growth on glucose [24], which are included in the second scenario described in the previous section, were adapted to this model. The adaptations that were performed include disabling the reactions known to be inactive for this scenario (transferred from iAF1260 scenario 2) and setting the ratio between proton-translocating and non-proton-translocating NADH dehydrogenase to 1:1. The latter was implemented by combining the two NADH dehydrogenase reactions, one translocating 3 protons per molecule of NADH and the other not translocating any protons, into one reaction that translocates 1.5 protons per molecule of NADH. The uptake of any other carbon source was disallowed, and ammonium, phosphate, and sulfate were used as sole sources of nitrogen, phosphate, and sulfur, respectively.

As simulations with the set of bounds described by the above scenario displayed some discrepancies between model predictions and experimental data, an **alternative scenario** was employed in an attempt to resolve these discrepancies. This scenario was derived from the original scenario described above by adding a forced flux of

5 mmol gDW^{-1} h^{-1} through the following reaction, which is catalyzed by proton-translocating transhydrogenase (EC 1.6.1.2):

$$1\ NADH + 1\ NADP^+ + 2\ H^+[periplasm] \rightarrow 2\ H^+ + 1\ NAD^+ + 1\ NADPH$$

(unmarked metabolites represent cytoplasmic pools).

3.1.2 *Thermotoga maritima* DSM8: Model iTZ478

A whole-genome metabolic reconstruction of *T. maritima*, designated iTZ478, was published in 2009 [155]. To the author's knowledge, it remains the most current metabolic model of *T. maritima*. The model was downloaded from the supplementary material of the publication [155] in the SBML format. It covers 478 metabolic genes and encompasses two compartments (cytoplasmic and extracellular), 645 reactions (562 metabolic or transport reactions, 82 exchange reactions, and 1 biomass reaction), and 564 metabolite nodes (491 cytoplasmic and 73 extracellular). It employs a GAM of 45.56 mmol ATP gDW^{-1}. The model does not include an ATP hydrolysis reaction for modeling the energy requirement of non-growth-associated processes.

Owing to the lack of experimental data on the biomass composition of *T. maritima*, the modelers adapted the biomass reaction from an earlier metabolic reconstruction of *E. coli*, iJR904 [77], the predecessor of iAF1260, with the following modifications: The specific peptidoglycan composition of *T. maritima* is represented in the model, and specific lipids were incorporated, as far as quantitative information was available. The proteinogenic amino acid L-asparagine does not occur in the model. As its biosynthesis in *T. maritima* is thought to occur in a tRNA-bound state [156] and tRNAs were judged by the authors of iTZ478 to lie outside the scope of the model, L-asparagine was replaced with L-aspartate in the biomass reaction. The authors state that the model is not suitable for quantitative predictions, but should be able to make qualitatively correct predictions on gene essentiality and growth capability on different carbon sources [155].

In the studied scenario (defined by the lower and upper flux bounds present in the SBML file), glucose uptake is limited to a rate of 10 mmol gDW^{-1} h^{-1}. This value does not seem to be based on experimental measurements. The sole nitrogen source is ammonium, the source of phosphorus is phosphate, and sulfur is imported exclusively in the form of elemental sulfur. All of these can be taken up at arbitrary rates. In the model, no distinction is made between ammonia and ammonium.

3.1.3 *Pseudomonas aeruginosa* PAO1: Model iMO1056

The model iMO1056 was published in 2008 [79]. It contains 874 metabolite nodes and 883 reactions (1,005 including artificial biomass and exchange reactions), accounting for 1,056 genes. The model was downloaded from the supplementary material of the corresponding publication [79]. The minimal medium constraints supplied in the same file as the model (aerobic growth on glucose, glucose uptake rate: 10 mmol gDW^{-1} h^{-1})

were used for all simulations. In this scenario, the sole nitrogen source is ammonium, phosphorus is taken up in the form of phosphate, and the sole sulfur source is sulfate.

3.2 Flux balance analysis (FBA)

Flux balance analysis (FBA) is a mathematical method for predicting the fluxes in a stoichiometric metabolic network in steady state [15–17]. The key assumption underlying FBA is that the organism under consideration has been optimized through evolution for a biological goal, usually maximum growth at optimal utilization of available resources. FBA thus solves the following mathematical optimization problem:

Algorithm 1 (Flux balance analysis):

Maximize or minimize an objective function $Z(v)$

in steady state: $S \cdot v = 0$

with linear inequality constraints: $\text{lb} \leq v \leq \text{ub}$.

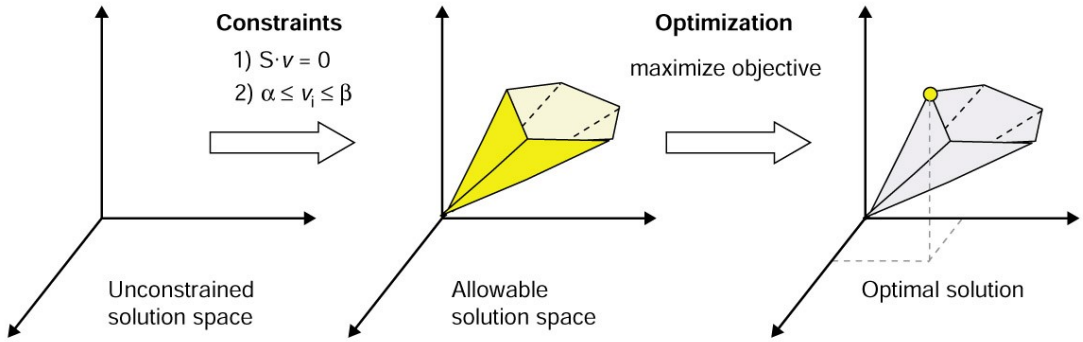

Figure 6: Principle of flux balance analysis: FBA uses mathematical optimization to select a particular solution from an allowable solution space defined by a set of linear equality and inequality constraints. Figure taken from [77].

The most commonly used objective function is maximization of biomass production, which corresponds to the flux through the biomass reaction in the model (see Section 2.3.5):

$$Z(v) = v_{\text{BM}}. \tag{9}$$

This is a linear function in **v**, i.e. a special case of

$$Z(v) = c^{\text{T}} v, \tag{10}$$

where **c** is an $n \times 1$ coefficient vector.

In this case, the optimization problem can be solved using linear programming [157], as the objective function and all equality and inequality constraints are linear in **v**. Figure 6 illustrates the principle of FBA schematically.

Other objective functions that have been examined include maximization of ATP yield, biomass yield per flux unit, or ATP yield per flux unit and minimization of total flux or glucose consumption [158]. Some of these, such as ATP yield per flux unit, are nonlinear functions, which require nonlinear programming. In contrast to nonlinear programming, linear programming is always guaranteed to find a global optimum of the objective function [157].

The reaction fluxes predicted by FBA are net fluxes. Hence, a flux of zero does not necessarily mean that the reaction is inactive, but merely that the effective rates of the forward and reverse reactions cancel each other out.

It should further be noted that the solution of the optimization problem is usually not unique and that out of all optimal solutions, FBA will return an arbitrary one. Seemingly insignificant changes in the input, such as changing the order of the reactions in the model, can lead to a different, but also optimal solution. The subregion of the solution space where the objective function assumes the optimum will be referred to as optimal solution space from this point on.

3.3 Dead-end analysis

A metabolic *dead end* is a metabolite that cannot be consumed and produced via separate reactions in the network [77]. There are two classes of dead ends, those for which no consuming or no producing reactions exist and those involved in only a single reversible reaction. In steady state, where the consuming and producing fluxes of each metabolite have to cancel each other out, all reactions that involve a dead-end metabolite necessarily have a flux of zero. The reactions that are blocked due to dead ends can, in turn, induce further dead ends. If, e.g., a reaction producing a dead-end metabolite is also the only reaction to consume another metabolite, that metabolite is a dead end as well. This can result in cascades of blocked reactions and induced dead ends [159]. Dead ends (primary or induced) can be identified exhaustively by the following iterative algorithm:

Algorithm 2 (Iterative dead-end analysis):

```
Preprocessing: Compile sets producingReactions[i],
consumingReactions[i] for each metabolite i
deadEnds := ∅
repeat
  newDeadEnds := ∅
  for each metabolite i not in deadEnds
    if (producingReactions[i] = ∅ or consumingReactions[i] = ∅ or
        | producingReactions[i] ∩ consumingReactions[i] | < 2)
```

```
        newDeadEnds := newDeadEnds U {i}
        Mark all reactions involving metabolite i as blocked
      end if
    end for
    deadEnds := deadEnds U newDeadEnds
    for each reaction j newly marked as blocked
      for each metabolite i not in deadEnds
        producingReactions[i] := producingReactions[i] \ {j}
        consumingReactions[i] := consumingReactions[i] \ {j}
      end for
    end for
  until newDeadEnds = ∅
```

In Algorithm 2, ∅ denotes the empty set; ∪, ∩, and \ are the usual set operators; and $|x|$ denotes the cardinality of set x.

Dead ends can indicate gaps or errors in the network [85]. As the medium composition is represented in the model in the form of restrictions (i.e. bounds) on transport or exchange reactions, dead ends occur regularly in degradation pathways for compounds that are not present in the medium.

For any analysis under steady-state conditions, all dead ends (primary or induced) can be removed from the network along with all reactions in which they are involved [160]. This reduces the dimension of the stoichiometric matrix **S** and thus the dimension of the solution space of the steady-state equation (Equation 2; see Section 2.3.1), which can greatly reduce the running times of algorithms operating in that solution space.

3.4 Flux variability analysis (FVA)

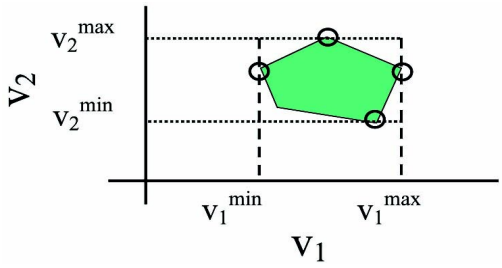

Figure 7: Flux variability analysis determines a pair of bounds $\left(v_i^{\min}, v_i^{\max}\right)$ *for each flux variable* v_i *of the network within which a (sub)maximal biomass flux is possible. Figure taken from [162].*

When using a linear objective function, FBA is guaranteed to find an optimal solution if the solution space is not empty. However, this solution is not necessarily unique [161]. In genome-scale networks, there usually exist alternative pathways, which are equivalent with respect to the objective function. Flux variability analysis (FVA) is a method for exploring the shape of the optimal solution space [162]. It is only applicable

to linear problems. FVA determines minimum and maximum values for each flux variable under the added constraint that the objective function of the FBA be optimal or near optimal. In geometric terms, FVA finds a hyperrectangle (box) in *n*-dimensional space within which all (sub)optimal solutions lie. This is illustrated for two dimensions in Figure 7.

The difference between the maximum and minimum values for each flux is called flux variability. It is a measure for how strongly a flux is restricted by the additional constraint of (sub)optimal biomass flux. Fluxes with low variability are essential for the formation of biomass, while highly variable fluxes can be replaced by other fluxes in the network.

FVA consecutively solves two linear optimization problems per reaction in the network. This is described more formally below.

Algorithm 3 (Flux variability analysis):

For each reaction j:

Minimize v_j

in steady state: $S \cdot v = 0$

with linear inequality constraints: $\text{lb} \leq v \leq \text{ub}$

and the additional constraint $|Z(v)| \geq \gamma \cdot |Z_{\text{opt, FBA}}|$.

Repeat the above with maximization instead of minimization (or, equivalently, minimization of $-v_j$ instead of v_j).

In Algorithm 3, $Z(\mathbf{v})$ denotes the objective function of the FBA (usually biomass flux), $Z_{\text{opt, FBA}}$ is the FBA optimum, and γ is a threshold parameter between zero and one. This parameter determines how much the optimality constraint is to be relaxed (as a fraction of $Z_{\text{opt, FBA}}$). For instance, a value of 0.95 signifies that the objective function value may differ by up to 5% from the optimum. If γ is set to one, only truly optimal solutions are explored.

It is worth noting that a solution exists for each linear optimization problem considered in FVA, because any FBA solution lies within the solution space. The running time of FVA can be improved in practice by launching each optimization from the previous optimal solution instead of starting from a fix or random point in the solution space or, worse, starting over every time by generating an initial feasible solution [163]. This improvement is based on the observation that the solutions of linear optimization problems with slightly changed objective functions tend to be located near each other in the search space [163].

In the interpretation of FVA results it is important to note that a very large or infinite flux variability does not usually signify the presence of true biological alternatives,

where flux can be distributed arbitrarily between alternative metabolic routes. Indeed, a state where two pathways with exactly the same function are present would likely be unstable over time. Without strong conservative selection, one of the two pathways would be free to degenerate or to evolve a new function, as is the case with duplicated genes [164,165]. Rather, infinite variability is an indicator of uncertainty in the model, e.g. regarding cofactor specificity or reaction directionality [103]. Moreover, the duplication of any reaction with unbounded flux in the model results in two reactions with infinite flux variability.

3.5 Minimization of metabolic adjustment (MOMA)

The core assumption of FBA, that the studied organism has been optimized through evolution for optimal growth (or a different biological goal), is only valid for wild-type organisms growing in natural or near-natural conditions. Artificially created mutants have not had the time to undergo evolutionary optimization. Major rerouting of metabolic fluxes, e.g. in response to the knockout of a metabolic enzyme, is likely impeded by regulatory mechanisms, including gene regulation and enzyme inhibition, which are still the same as in the wild type. Similarly, the evolutionary assumption is not valid for the wild type growing in a medium with a very different composition from the organism's natural habitat or in other non-natural conditions.

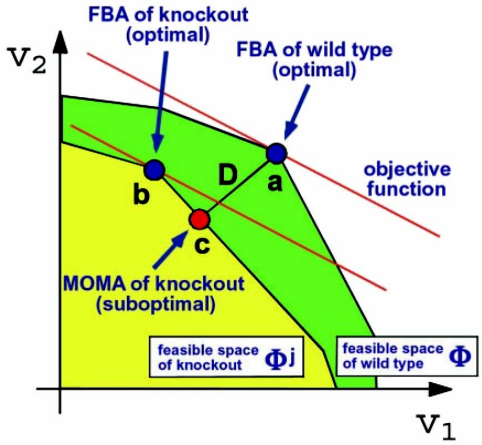

Figure 8: Minimization of metabolic adjustment (MOMA) projects the flux distribution of the wild type (**a**) *onto the allowable solution space of the mutant (yellow). The solution* (**c**) *can differ strongly from the FBA solution of the mutant* (**b**)*. Figure taken from [166].*

Minimization of metabolic adjustment (MOMA) is based instead on the assumption that the flux distribution in a perturbed network will be as close to that of the unperturbed (wild-type) network as is still possible within the modified solution space [166]. It uses the same equality and inequality constraints as FBA (aside from constraints defining the genetic or environmental modification) but a different objective function. While FBA maximizes the biomass flux, MOMA minimizes the Euclidean distance of the flux vector to the flux distribution of the unperturbed network. Geometrically, this

corresponds to a projection of the wild-type flux vector onto the solution space of the perturbed network, as shown schematically in Figure 8.

Given a reference flux distribution $\mathbf{v}_{WT}$, MOMA solves the following optimization problem:

Algorithm 4 (Minimization of metabolic adjustment):

Minimize $D(v) = (v - v_{WT})^T (v - v_{WT})$

in steady state: $S \cdot v = 0$

with linear inequality constraints: $\text{lb} \leq v \leq \text{ub}$.

In practice, the square of the Euclidean distance is used as objective function (see Algorithm 4). This is possible because squaring is a monotonically increasing function, so that minimizing the original or the squared objective function will yield the same solution [166]. Similarly, any constant offset or factor can be omitted from the objective function in mathematical optimization. The squared objective function is quadratic in **v**. Hence, the optimization problem can be solved using convex quadratic programming [167].

While FBA allows the prediction of network behavior from network topology, network stoichiometry, and a set of constraints on the fluxes alone, MOMA requires a reference flux distribution. This limitation is shared by regulatory on/off minimization (ROOM), a related method for predicting the flux distribution of perturbed networks. In contrast to MOMA, ROOM minimizes the number of changed fluxes rather than the total amount of flux change compared to a reference flux vector [168]. In both methods, an FBA solution for the unperturbed network is usually employed as reference.

3.6 Weighted MOMA (wMOMA)

The solutions returned by MOMA and related methods are only as good as the reference vector. Using an experimentally determined flux distribution would be ideal. However, while ^{13}C tracer experiments can resolve the high-magnitude fluxes in central carbon metabolism and in amino acid biosynthesis pathways [119,169], it is impossible to determine the *in-vivo* fluxes through all reactions in the network. Because of this limitation, *in-silico* predictions obtained via FBA are used instead.

If the metabolism of the modeled organism is not well understood, a high degree of uncertainty may remain in the use of cofactors and alternative reaction sets. Hence, the difference between extreme FBA solutions from opposite ends of the optimal solution space may be substantial. MOMA predictions for a perturbed network can consequently differ widely depending on the FBA solution used as reference. Weighted MOMA, a method developed in the present work, uses a weighted Euclidean distance as objective

function, with weights based on the degree of uncertainty in the FBA solution. The degree of uncertainty for each flux variable is assessed via FVA.

Weighted MOMA thus performs the following optimization, based on the given reference flux distribution $\mathbf{v}_{WT}$:

Algorithm 5 (Weighted MOMA):

Perform FVA for wild type $\rightarrow v_i^{\min}, v_i^{\max}$.

Compute weights from flux variability:

$$w_i = \alpha + \exp\left(-\beta\left(v_i^{\max} - v_i^{\min}\right)\right).$$

Minimize $D(v) = (v - v_{WT})^T \operatorname{diag}(w)(v - v_{WT})$

in steady state: $S \cdot v = 0$

with linear inequality constraints: $\mathrm{lb} \leq v \leq \mathrm{ub}$.

In Algorithm 5, **w** is an $n \times 1$ vector of weights. The optimality parameter γ for FVA is an additional input of the algorithm, as are the values of the positive parameters α and β.

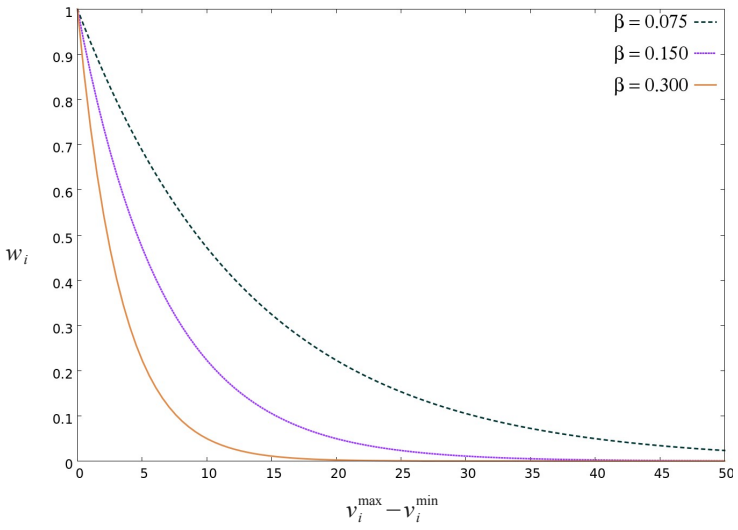

Figure 9: The exponentially decreasing weight function used in weighted MOMA: Weight w_i as function of the variability of flux v_i. Flux variability is determined via FVA. A value of $\alpha = 10^{-6}$ was used for all three curves.

The decreasing exponential function that was chosen for the computation of the weights assigns a weight of one to fluxes that are restricted to a single value by the constraint of (sub)optimal biomass. This function was chosen because it is most sensitive in the relevant region near zero and assigns a very low weight (close to the base value of α) to

highly variable fluxes. The parameter β determines the steepness of the function. Figure 9 shows the weight function for three different values of β.

This weight function ensures that fluxes that are sharply restricted by the constraint of optimal biomass are weighted up, while unrestricted fluxes are weighted down.

3.7 Split-ratio analysis

The following algorithm was developed in the course of this work for the computation of metabolite fluxes and split ratios from reaction fluxes. It was first described in [170].

Algorithm 6 (Split-ratio analysis):

Given a stoichiometric matrix **S** and a flux vector **v**,

For each metabolite *i*:

Compute partial fluxes $\rho_{ij} = s_{ij} \cdot v_j$ for all reactions *j* and group into positive (producing) and negative (consuming):

$P_i = \{j \mid \rho_{ij} > 0\}, \quad C_i = \{j \mid \rho_{ij} < 0\}.$

Compute $\Phi_i = \sum_{j \in P_i} \rho_{ij} = -\sum_{j \in C_i} \rho_{ij}$ and divide partial fluxes by this sum:

$$\rho_{ij}^* = \frac{\rho_{ij}}{\Phi_i}.$$

The ρ_{ij}^* computed by Algorithm 6 are collectively called the split ratios of metabolite *i*. If positive, ρ_{ij}^* is the fraction of metabolite *i* produced via reaction *j*. If negative, it is the fraction consumed via that reaction. The ρ_{ij}^* are usually expressed as percentages of Φ_i, the total flux through metabolite *i*. Φ_i is also called the flux sum of metabolite *i* [11].

Based on split ratio analysis, metabolite nodes can be divided into three classes for the growth scenario under consideration: Inactive metabolite nodes, intermediates, and branch points. A metabolite is inactive if it does not occur in any reaction with a non-zero flux, which results in a flux sum of zero. A pure intermediate or transitory metabolite node is a metabolite that is produced by exactly one reaction and consumed by exactly one reaction. Branch points are metabolite nodes where fluxes branch or join, i.e. metabolites involved in more than two active reactions.

3.8 Branch point analysis (BPA)

Branch point analysis (BPA) is a graphical extension of split-ratio analysis, which allows the visual analysis of the extended metabolic context of a selected metabolite or reaction or sets of metabolites and reactions. The method was developed in our group by René Rex [170].

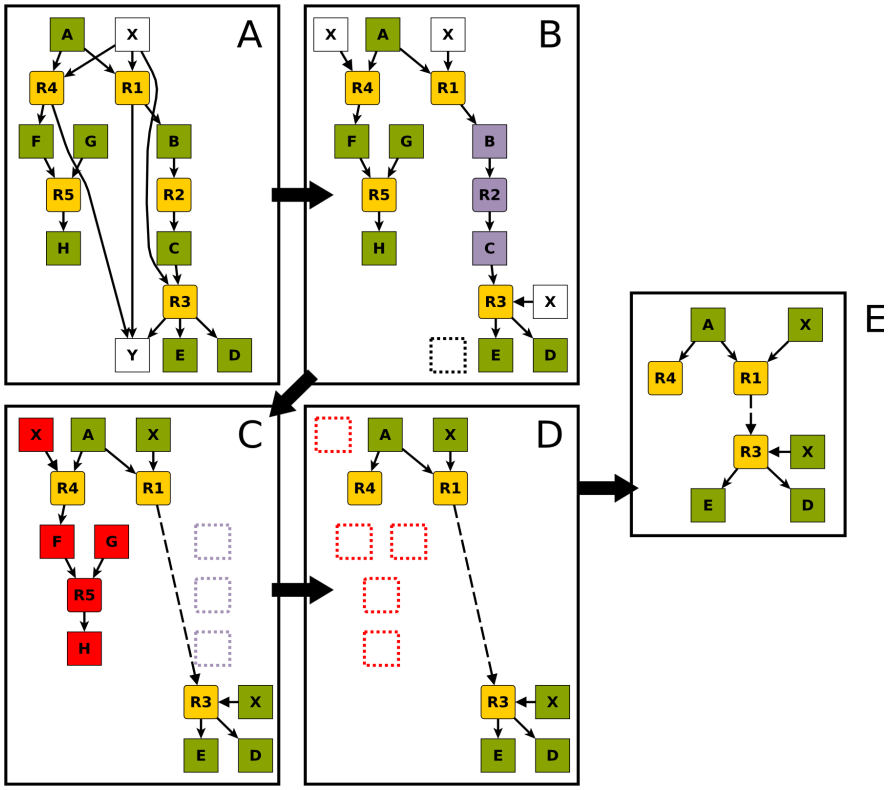

Figure 10: The algorithm of branch point analysis proceeds in four steps: (A) The full bipartite graph is generated via split-ratio analysis. In the example, R1 *has been selected as hub,* X *is to be disconnected, and* Y *is to be removed. (B) Selected nodes are disconnected* (X) *and removed* (Y)*, respectively. (C) Each sequence of transitory nodes* (–B–R2–C–) *is bridged by a combined edge. (D) Nodes with a distance from the hub of more than the selected maximum depth (2) are removed. (E) Output graph. Figure taken from [170].*

Figure 10 illustrates the algorithm for a small example network. In the simplest case, the inputs of the algorithm are a flux distribution (e.g. an FBA solution), a hub node, a list of nodes (reactions or metabolites) to be excluded, a list of nodes to be disconnected, and a parameter *maxDepth*. The hub is the reaction or metabolite node whose metabolic context is to be displayed. By excluding or disconnecting certain nodes, the graph output can be made clearer. Typical candidates for exclusion are metabolites that are ubiquitous but not of interest in most conditions, such as H^+, H_2O, and phosphate. Other promiscuous metabolites, i.e. metabolites participating in a large number of active reactions, such as ATP, NADH, and coenzyme A, should be disconnected in order to prevent crowding of the graph.

BPA overlays the split ratios and metabolite fluxes computed by split-ratio analysis on a bipartite graph representation of the metabolic network (see Section 2.3) reduced to the selected metabolic context. The algorithm is described in detail below.

Algorithm 7 (Branch point analysis):

Perform split-ratio analysis on the given flux distribution.

Construct the bipartite graph from the active reactions and metabolites; set edge directions from the signs of the reaction fluxes.

Label edges with split ratios (as percentage of the flux through the adjacent metabolite node) and metabolite nodes with the respective flux sums.

Remove each node to be excluded from the graph, along with the adjacent edges.

Replace each node to be disconnected by separate instances for each edge involving the node.

While the graph contains nodes with exactly one incoming and one outgoing edge (transitory nodes):

Pick a transitory node T. Let P→T and T→N denote the incoming and outgoing edges, respectively.

Remove T, P→T, and T→N from the network.

Introduce a new combined edge P→N.

Remove all nodes from the graph that have a distance of more than *maxDepth* from the hub node.

Note that the final graph contains only metabolic branch points, i.e. nodes where fluxes join or split, except for the hub, which is included in the graph even if it is a transitory node.

By selecting more than one hub for the analysis, it is possible to use BPA for the visual analysis of larger metabolic contexts, such as metabolic pathways or metabolite pools. In our implementation, only the largest connected component is displayed in case the resulting graph is disconnected.

BPA was implemented by René Rex as an optional extension of *metano*, the software toolkit presented in this thesis. The resulting program, AMEBA (Advanced MEtabolic Branchpoint Analysis), can be downloaded from http://metano.tu-bs.de/ameba. Like *metano*, it is implemented in the Python programming language and distributed under the GNU General Public License version 3.

3.9 Metabolite flux minimization (MFM)

The following algorithm computes the minimum flux through each metabolite node under the added constraint that the objective function of the FBA be (sub)optimal. Like FVA, it is only applicable to linear problems. The algorithm was developed in the course of this work and first described in [170].

Metabolite flux minimization (MFM) operates on flux variables that are split into non-negative components:

$$v_j = v_j^+ - v_j^-, \text{ where } v_j^+ \geq 0 \text{ and } v_j^- \geq 0.$$

This splitting is achieved by adding negative copies of the columns of the stoichiometric matrix **S** to the matrix. Lower and upper bounds have to be adjusted accordingly. In practice, flux variables constrained to zero, e.g. because the original flux variable was constrained to be non-negative or non-positive, can be removed along with the corresponding columns of **S**.

Algorithm 8 (Metabolite flux minimization):

Transform the flux vector **v** into a vector of non-negative flux variables and adapt the stoichiometric matrix **S** and the bounds vectors **lb** and **ub** accordingly.

For each metabolite *i*:

Minimize $\sum_j s_{ij}^+ \cdot v_j$, where $s_{ij}^+ = \begin{cases} s_{ij}, s_{ij} \geq 0 \\ 0, \text{ else} \end{cases}$

in steady state: $S \cdot v = 0$

with linear inequality constraints: $\text{lb} \leq v \leq \text{ub}$

and the additional constraint $|Z(v)| \geq \gamma \cdot |Z_{\text{opt, FBA}}|$.

MFM solves one linear optimization problem per metabolite node in the network. In Algorithm 8, $Z(\mathbf{v})$, $Z_{\text{opt, FBA}}$, and γ have the same meanings as in FVA (see Section 3.4). γ is again a threshold parameter between zero and one determining how much the optimality constraint is to be relaxed. The algorithm successively minimizes the producing flux for each metabolite node.

3.10 Automated plausibility checking of metabolic models

When reconstructing the metabolic model of a particular organism, certain metabolic behaviors will be expected, e.g. based on the organism's taxonomy and ecology. For instance, in an organism known to be strictly aerobic, the model should not allow for growth without uptake of oxygen. To give another example, in all organisms in which

both glycolysis and gluconeogenesis are present, a strong glycolytic flux is to be expected for growth on carbohydrates, while gluconeogenesis can be expected to be active for most other substrates that can be utilized.

Such expectations can be used to automatically check a metabolic model for biological plausibility in a given growth scenario or set of scenarios. Any such expectation can be formalized as a set of assertions, i.e. expressions that can be evaluated to `True` or `False`, where the expected result evaluates to `True` [171]. The assertions contain variables representing, e.g., reaction fluxes, metabolite fluxes, split ratios, and the minimum and maximum values for reaction fluxes determined by FVA.

The following algorithm automatically processes a list of assertions and reports those that are violated in the scenario under consideration:

Algorithm 9 (Evaluation of model assertions):

Given a reaction file, scenario file, FVA threshold parameter, and list of assertions:

Perform FBA

Compute metabolite fluxes and split ratios from FBA solution

Perform FVA with the given threshold parameter

Evaluate all assertions

Report all assertions that failed or could not be evaluated

3.11 Prediction of reaction directionality from Gibbs free energies

In principle, all chemical reactions are reversible [172]. Under physiological conditions, however, a reaction may very well be irreversible, depending on the change in Gibbs free energy associated with that reaction. For any reaction *j*, the difference in Gibbs free energy between reactants and products can be computed from the Gibbs energies of formation of each reactant and product as follows:

$$\Delta_r G_j = \sum_i s_{ij} \Delta_f G_i^0 + RT \ln\left(\prod_i c_i^{s_{ij}}\right) \text{ [101].} \qquad (11)$$

In Equation 11, s_{ij} is the stoichiometric coefficient of metabolite *i* in reaction *j* as defined in Section 2.3.1, c_i is the concentration of metabolite *i*, $\Delta_f G_i^0$ is the standard Gibbs energy of formation of metabolite *i*, *T* is the absolute temperature in Kelvin, and *R* is the gas constant (≈ 8.314 J mol^{-1} K^{-1}). The (transformed) standard Gibbs energies of formation $\Delta_f G_i^0$ are approximately constant for given ionic strength and pH values [173]. They have been experimentally determined [174] or estimated using group contribution methods [175] for many biomolecules [176].

The net reaction proceeds from left to right if $\Delta_r G_j$ is negative, else from right to left. It is obvious from Equation 11 that the direction of a biochemical reaction depends only on the temperature and the concentrations of the reactants and products. As the temperature is constant in steady state, the directionality of biochemical reactions can be predicted from the physiological ranges of the metabolite concentrations [101,113]:

A lower bound for $\Delta_r G_j$ can be computed via Equation 11 by setting the concentrations of the products (which have positive coefficients s_{ij}) to their lower bounds and those of the reactants (which have negative coefficients s_{ij}) to their upper bounds. Similarly, an upper bound is obtained by using the upper bounds of the product concentrations and the lower bounds of the reactant concentrations.

If the lower and upper bounds of $\Delta_r G_j$ computed in this way have the same sign, the reaction is irreversible for the given physiological concentration ranges. If the sign is negative, the reaction proceeds in the forward direction, else in the backward direction. If the lower and upper bounds have opposite signs, the reaction is reversible.

Obviously, if the measured physiological ranges for the metabolite concentrations are too broad, nearly all reactions will be classified as reversible, which reduces the utility of this approach. If, on the other hand, the measured ranges underestimate the actual variability, reactions may falsely be assumed to be irreversible, which may exclude physiological metabolic capabilities.

4 Implementation and validation

4.1 Overview

metano is a collection of algorithmic tools for analyzing the capabilities of metabolic networks and for assisting in metabolic reconstruction. It has been implemented in the Python programming language[1] and is designed to be easy to use and run reasonably fast. *metano* is distributed as open source under the GNU General Public License version 3 (http://www.gnu.org/licenses/gpl.html). Source code and documentation are provided via the website http://metano.tu-bs.de. An object-oriented approach has been chosen for the design and implementation in order to ensure long-term maintainability and easy extensibility of the software.

All tools in *metano* are available in two forms, as stand-alone programs and as Python classes. The stand-alone programs are designed to run in any Linux environment. They can be run from the Linux console and require neither a graphical display nor a database connection, which makes them well-suited for batch execution in a massively parallel computing environment. The stand-alone scripts use plain ASCII files for input and output. The scripts require no adjustment of numerical parameters so as to be easy to use by metabolic modelers with little or no programming experience. On the other hand, the Python classes and functions of *metano* have been designed in such a way that they can be directly included in any Python software project. The interfaces (constructors and parameter lists) have been designed with the goal of easy reuse, requiring as little context as possible.

The computational methods implemented in *metano* fall into three categories: (i) Prediction of flux distributions and reaction-centric analyses (FBA, FVA, MOMA, wMOMA), (ii) assistance in the construction of metabolic models (dead-end analysis, automated plausibility checking, prediction of reaction directionality), and (iii) metabolite-centric analyses (split-ratio analysis, metabolite flux minimization). In addition, *metano* features parsers and writers for both its native file formats and the Systems Biology Markup Language (SBML), a standard for the exchange of biological network models [153,177].

Algorithms with a long running time have been parallelized using the Message-Passing Interface (MPI) [178] via the mpi4py interface[2], so that they can be run on cluster computers without modification of the code.

metano includes interfaces to a number of solvers for mathematical optimization problems, most of which are also open source software: For linear programming, the

1 Python Software Foundation, http://www.python.org

2 MPI for Python, L. Dalcin, http://mpi4py.scipy.org

supported solvers are lp_solve[1] and the GNU Linear Programming Kit (GLPK)[2], interfaced via OpenOpt[3] and PyMathProg[4], respectively, which provide Python bindings. For quadratic programming, *metano* uses the quadratic cone program solver implemented in CVXOPT[5], accessed via the CVXMOD modeling interface[6]. For non-linear programming, the ALGENCAN solver [179] is used, which is available free of charge for academic users[7]. ALGENCAN is also accessed via the OpenOpt interface.

4.2 Class structure

The design of *metano* follows the object-oriented paradigm of software development [180]. The classes of *metano* fall into three categories: classes for data storage and exchange, input/output classes, and algorithmic classes.

The central data storage classes of *metano* are MetabolicModel and MetabolicFlux. The first is used to represent a metabolic model, while the latter is used for the representation of a flux distribution in a network, i.e. a flux vector. A MetabolicModel encompasses not only the network itself, but also network-associated parameters such as flux bounds. The MetabolicModel class stores the biochemical reactions forming the network as a list of objects of the class Reaction. The Reaction class stores the reaction name, the reaction equation in the form of lists of (coefficient, metabolite) pairs for the two sides of the reaction, and the lower and upper bounds for the flux through the reaction. The special value `inf` is used to denote the absence of a bound (`+inf` for upper and `-inf` for lower bounds). The (coefficient, metabolite) pairs are stored as instances of the class Reactant instead of simple tuples, which allows addressing the values by name rather than by index. The MetabolicModel class was designed in cooperation with René Rex.

The member functions of the MetabolicModel class include functions for filling a MetabolicModel with data read from files (via the ReactionParser and ParamParser classes described below) and for writing a metabolic model to ASCII files in *metano*'s native reaction file format. Furthermore, the class provides functions for extracting sub-networks (also represented as instances of MetabolicModel) according to user-provided criteria and for constructing the stoichiometric matrix from the list of reactions. Finally, MetabolicModel implements topology-based analyses, including the automatic detection of boundary reactions and the identification of duplicate reactions in the network. As the presence of dead ends is a network property, dead-end analysis (see

1 lp_solve 5.5, http://lpsolve.sourceforge.net

2 The GNU project, http://www.gnu.org/software/glpk

3 National Academy of Sciences of Ukraine, http://www.openopt.org

4 http://pymprog.sourceforge.net

5 M. Andersen, J. Dahl, and L. Vandenberghe, http://abel.ee.ucla.edu/cvxopt

6 J. Mattingley, Stanford University, http://cvxmod.net

7 TANGO project, http://www.ime.usp.br/~egbirgin/tango/downloads.php

Section 3.3) has also been implemented as a member function of the MetabolicModel class.

The MetabolicFlux class represents a flux distribution as a Python dictionary assigning a flux value to each reaction by name. In addition, it can optionally store the effective flux bounds in a similar dictionary. It provides functions for reading and writing flux distributions in *metano*'s native ASCII file format and for the conversion between different representations of flux distributions. The most important of these conversions are from and to a flat vector, represented as a Python list, where flux values are addressed by index rather than by reaction name. In these conversion functions, a MetabolicModel can be supplied for providing the association between reaction names and indices. In addition, the MetabolicFlux class implements different semimetrics for computing the distance between two flux vectors (currently Manhattan distance and squared Euclidean distance). Finally, split-ratio analysis (see Section 3.7) has been implemented as a member function of the MetabolicFlux class.

The dedicated input/output classes are ReactionParser, ParamParser, SbmlParser, and SbmlWriter. *metano* natively uses plain ASCII formats for storing lists of reactions (reaction files), scenario-specific parameters (scenario files), and flux distributions (flux files). Detailed descriptions of these file formats can be found online under http://metano.tu-bs.de/quickref.html#fileform. A parser for reaction files has been implemented in the class ReactionParser. As mentioned above, instances of MetabolicModel can be constructed directly from reaction files. The corresponding function internally calls a ReactionParser to generate a list of Reaction objects. Similarly, the ParamParser is used for parsing a scenario file, which most importantly contains explicit definitions of flux bounds to be applied in addition to the implicit bounds imposed by the irreversibility of reactions. The resulting lists of explicit (finite) bounds can then be used to update the bounds stored in a MetabolicModel in order to adapt the MetabolicModel to the simulation scenario.

Besides its native file formats, *metano* also supports the SBML format (Systems Biology Markup Language). The classes SbmlParser and SbmlWriter implement a parser and an export filter, respectively, for SBML files. The two classes and the associated stand-alone Python scripts for the conversion between the file formats of *metano* and SBML were originally developed by Alexander Krause in the course of a 3-week internship under supervision of the present author. All elements represented in one MetabolicModel and one MetabolicFlux object can be represented in one SBML file. However, owing to the great flexibility of the language, most elements in SBML are optional. Hence, an SBML file from a third party will usually not contain all elements necessary for constructing a MetabolicModel or for performing flux balance analysis on the model.

The most important algorithmic classes of *metano* are FbAnalyzer, FvAnalyzer, MomaAnalyzer, MfmAnalyzer, and ModelWatcher. They implement FBA (see Section 3.2), FVA (see Section 3.4), MOMA/wMOMA (see Sections 3.5 and 3.6), metabolite flux minimization (see Section 3.9), and the automated plausibility checking described in Section 3.10, respectively. FbAnalyzer, FvAnalyzer, MomaAnalyzer, and

MfmAnalyzer internally use the following classes for representing and solving the mathematical optimization problems arising in the analyses: LinearProblem, QuadraticProblem, MomaProblem, and NonLinearProblem. The inheritance schema of these classes is shown in Figure 11.

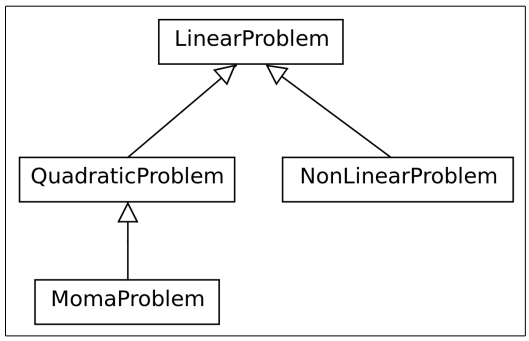

Figure 11: Inheritance schema for the metano *classes for representing and solving mathematical optimization problems.*

LinearProblem is used to represent linear programs, QuadraticProblem represents quadratic programs, and NonLinearProblem represents general nonlinear programs. MomaProblem is a specialization of QuadraticProblem for representing and efficiently solving the specific problem occurring in MOMA, i.e. minimization of Euclidean distance, which is a convex quadratic program.

4.3 Flux balance analysis

Virtually all existing implementations of FBA assign arbitrary large lower and upper bounds (e.g. −10,000 and +10,000) to all flux variables for which no biological restrictions are known [14,103,181–183]. In these implementations, each flux variable v_i has a finite lower bound lb_i and a finite upper bound ub_i. A notable exception is the recently-published CoPE-FBA [21]. The introduction of artificial bounds amounts to a modification of the solution space: The original flux cone, which is unbounded in some dimensions, is replaced by a polytope, which is bounded in all dimensions [21]. This modification introduces additional vertices to the optimal solution space. This is illustrated schematically in Figure 12.

Linear optimization problems such as those occurring in FBA are usually solved using a variant of the simplex method [94,184]. The simplex method solves a linear program on a solution space defined by sets of equality and inequality constraints by finding a basis of the solution space (i.e. an initial solution) and subsequently performing a series of basis transformations [157]. In each iteration, all optimization variables that are not part of the basis are set to extreme values defined by the given inequality constraints (either the lower or the upper bound). Consequently, the simplex method only considers vertices of the solution space when looking for an optimal solution. As the simplex algorithm operates on non-negative variables, zero is assigned as default value to all non-basis variables that are not otherwise constrained. In FBA, the non-negative

optimization variables are obtained from the flux variables by splitting the latter into non-negative components (as described in Section 3.9). This is equivalent to modeling the forward and backward fluxes through each reaction by separate non-negative variables.

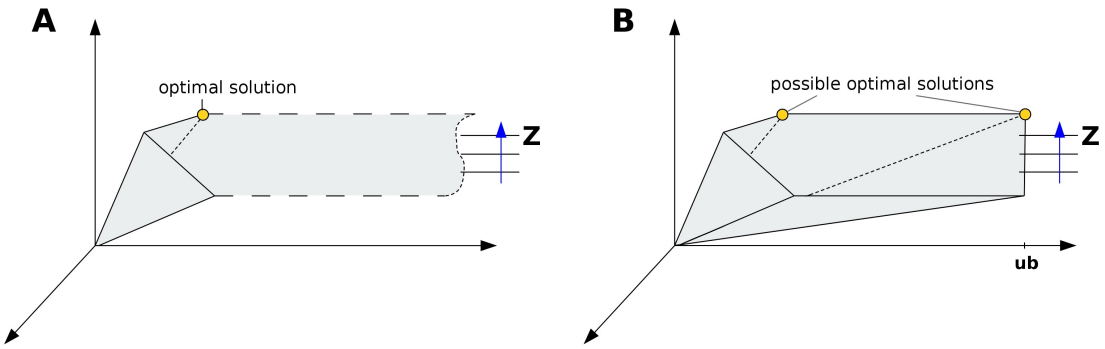

*Figure 12: The introduction of arbitrary large bounds on reaction fluxes with no known biological restriction (exemplified by upper bound **ub** in panel **B**) introduces spurious vertices to the solution space of flux balance analysis. These vertices can appear as possible solutions when solving the linear program with a variation of the simplex algorithm. Shown in gray is the solution space of FBA (**A**) without and (**B**) with an artificial large bound **ub**. **Z** is the (linear) objective function.*

As all optimal solutions returned by the simplex algorithm are vertices of the solution space, introducing artificial large bounds and thus additional vertices increases the number of possible optimal solutions (see Figure 12). The additional vertices correspond to flux distributions in which a subset of the artificial, non-biological large bounds is actually assumed. In this way, arbitrary large cyclic fluxes are assigned to reactions in thermodynamically infeasible loops, as such fluxes do not affect the value of the objective function (see Section 2.4). An example can be seen in the flux distribution predicted by the object-oriented open-source implementation SurreyFBA [185] for the *E. coli* model iAF1260 under scenario 1 (see Section 3.1.1.1). Table 21 in Appendix A shows the relevant excerpt from this flux distribution. The flux distribution contains 19 fluxes with an absolute value of more than 999,000 mmol gDW^{-1} h^{-1}.

As mentioned in Section 2.4, several widely used implementations of FBA employ artificial large bounds when formulating the linear program and remedy the problem of cyclic fluxes through infeasible loops by secondary corrections. These often involve nested optimization strategies or the exhaustive identification and elimination of infeasible loops. Among these implementations is the COBRA toolbox [14,186], which is possibly the most popular software tool for performing FBA. In contrast, the *metano* implementation of FBA applies only explicit bounds and lower or upper bounds of zero implied by the irreversibility of reactions. *metano* thus allows flux variables that are unbounded on one or both sides. If the flux through the biomass reaction is used as objective function and fluxes with no known biological restrictions are left unbounded, non-productive fluxes (i.e. those not contributing to biomass formation) do not occur in FBA. In particular, cyclic fluxes through thermodynamically infeasible loops do not occur. While the exhaustive identification of stoichiometrically balanced cycles may be

a worthwhile goal in itself, special treatment of infeasible loops is not required when performing FBA on the unmodified solution space.

FBA has been implemented in *metano* with the following features: Firstly, flux variables are split into non-negative components, as described in Section 3.9. While the simplex method only works on non-negative variables, virtually all existing solvers for linear programs include a presolver, which automatically performs such preprocessing. However, the advantage of explicitly working with split flux variables is that the total flux through the network can be computed easily as the sum of all flux variables.

Secondly, the implementation of FBA in *metano* includes dead-end analysis as a preprocessing step, which is performed before flux variables are split. In this step, the stoichiometric matrix **S** is reduced by iterative removal of all dead ends and blocked reactions from the network. FBA is then performed on the resulting stoichiometric matrix with reduced dimension. The reactions removed in preprocessing are added again to the returned solution with a flux of zero. The effect of reducing the dimension of the problem before optimization is a considerable speedup. For instance, the model iAF1260 can be reduced from 2,382 reactions and 1,668 metabolites to 2,148 reactions and 1,452 metabolites, which amounts to a decrease in the number of entries of the stoichiometric matrix by 22%. It should be noted that if an inequality constraint exists forcing a blocked reaction to assume a non-zero flux, the problem is infeasible.

Thirdly, *metano* FBA supports alternative objective functions besides the biomass flux. Any linear or general nonlinear function can be used: A linear function can be specified by its coefficient vector, while a nonlinear function can be given directly as a Python function and, optionally, a second Python function for the gradient. As detailed in Section 4.1, *metano* includes interfaces to a number of linear and nonlinear solvers. Since it is generally impossible to identify the global optimum of an arbitrary nonlinear function or even to determine with certainty whether the function is bounded, nonlinear optimization is repeated a user-specified number of times from randomly chosen starting points. In this case, the optimum of all runs is returned along with the corresponding flux distribution. Of course, this is still not guaranteed to be the global optimum.

Finally, *metano* provides the option to secondarily minimize the total flux through the network, i.e. the sum of the absolute flux values. This secondary optimization has been implemented as a binary search for the lowest possible value of an additional upper bound on the sum of all non-negative flux variables that still leads to an optimal FBA solution.

To demonstrate the correctness of the present implementation, it was applied to the *E. coli* model iAF1260 for scenario 1 described in Section 3.1.1.1 (glucose uptake rate: 8 mmol gDW^{-1} h^{-1}, oxygen uptake rate: $\leq$ 18.5 mmol gDW^{-1} h^{-1}). The predicted flux distribution was compared to solutions generated by two existing implementations of FBA, the COBRA toolbox [14,186] and SurreyFBA [185]. The COBRA toolbox was chosen because of its high popularity in the metabolic modeling community (for instance, a Google Scholar search returns over 440 hits for the string “COBRA toolbox”

as of March 2013). Furthermore, the authors of iAF1260 performed their analyses on the model using the COBRA toolbox and published their FBA solutions as supplementary material with the original publication [24]. SurreyFBA was chosen because it shares the design features of being both object-oriented and open-source software with *metano*.

The FBA solution of *metano* was obtained using the linear solver GLPK with the objective of maximizing the biomass flux without secondary minimization of total flux. The solution conforms to all linear and nonlinear bounds within numerical error: No upper bounds were violated, and no lower bound was violated by more than 4.1×10^{-17} mmol gDW^{-1} h^{-1}. The average deviation from zero in the vector $\mathbf{S}\cdot\mathbf{v}$ (see Equation 2 on page 9) was below 8.8×10^{-17} mmol gDW^{-1} h^{-1}, with a maximum absolute deviation of 2.6×10^{-14} mmol gDW^{-1} h^{-1}. The solution is optimal, as the predicted biomass flux agrees with the solutions of the COBRA toolbox and SurreyFBA to within 10^{-6} h^{-1} (both the COBRA toolbox solution and the SurreyFBA solution are rounded to six digits). A detailed comparison of the three flux distributions is given in Appendix A. The three flux distributions are in agreement except for alternative pathway use.

It is worth noting that of the three implementations only SurreyFBA predicts large cyclic fluxes through infeasible loops, as the COBRA toolbox implements a secondary optimization strategy to prevent the occurrence of such fluxes. The strategy used by the most recent version of the COBRA toolbox has been termed *loopless FBA* [103]. It involves solving a mixed-integer linear optimization problem instead of a linear program, which is much more computationally intensive [103,186].

4.4 Flux variability analysis

metano implements the ‘fastFVA’ variant of flux variability analysis [163], where each optimal solution discovered is used as the starting point of the following optimization (see Section 3.4 for a brief description). The FBA optimum of the objective function can be determined in one of three ways: It can either be obtained from an FBA solution passed as a MetabolicFlux object or a flux file or be computed by performing an FBA prior to the analysis.

As in FBA, a dead-end analysis is performed as a preprocessing step in order to speed up the analysis. As any reaction blocked due to dead ends necessarily has a flux of zero, no optimization is required in this case. FVA has, however, been implemented in *metano* without the explicit splitting of flux variables that is performed prior to FBA, as that would make the formulation of the resulting optimization problems less clear without having any benefit.

In contrast to FBA, arbitrary large bounds of ±100,000 mmol gDW^{-1} h^{-1} are used in FVA in order to ensure that all of the optimization problems are bounded. By using a sufficiently large (i.e. non-biological) value for the artificial bounds, fluxes with infinite variability can be identified reliably from the solution alone. Otherwise, it would be necessary to query the internal status of the linear program solver to distinguish between

infeasible and unbounded problems and possible error states. This would be difficult, as the supported solvers each use different representations and data structures for their internal status.

4.5 MOMA and weighted MOMA

MOMA and wMOMA have been implemented in a common set of Python functions – the method is selected via an optional parameter: If a vector of weights is supplied, wMOMA is performed, else MOMA. MOMA and wMOMA use the CVXOPT quadratic cone program solver by default, but any quadratic or general nonlinear solver can be used.

Reducing the stoichiometric matrix prior to analysis is even more important for MOMA and wMOMA than for FBA, because quadratic cone programming is more computationally intensive than linear programming. By default, a dead-end analysis is performed for the identification of blocked reactions. Alternatively, a more exhaustive approach is available that involves FVA: Any reaction that is restricted to a flux value of zero with a flux variability of zero (determined at a relaxed threshold parameter, e.g. $\gamma = 0.9$) can be removed from the stoichiometric matrix. All metabolites that are no longer involved in any reaction can then be removed in a second step. This approach can potentially identify reactions that are not blocked due to dead ends but nevertheless cannot carry a flux. For instance, for the *E. coli* model iAF1260 in scenario 1 (see Section 3.1.1.1), exhaustive dead-end analysis identifies 234 of the 2,382 reactions as blocked and 216 of the 1,668 metabolites as dead ends, while FVA reveals that 844 reactions and 628 metabolite nodes can never carry a flux. In this case, the number of entries in the stoichiometric matrix can be reduced by 60% (compared to 22% when using dead-end analysis). This strategy is, however, only applicable when the solution space of the mutant is a subset of the wild-type solution space. If the mutant scenario has additional reactions or relaxed constraints, reactions that are blocked in the wild type are not necessarily blocked in the mutant as well.

As mentioned above, any vector of weights can be used in wMOMA. Per default, the weight vector is computed from a given FVA result according to the decreasing exponential function described in Section 3.6. If parameter values are not supplied, the default values of $\alpha = 10^{-6}$ and $\beta = 0.15$ are used.

To automatically evaluate the results of MOMA and wMOMA on genome-scale metabolic models, a script for *in-silico* knockout analysis has been developed. It successively applies MOMA or wMOMA to all scenarios where exactly one reaction is knocked out, i.e. has its flux constrained to zero. This analysis serves mostly testing and validation purposes, as it does not use a biological concept of knockouts. In reality, it is of course impossible to knock out individual biochemical reactions. The biological entities that can be knocked out are stretches of DNA, including genes, regulatory elements, clusters of genes, and whole plasmids or chromosomes. This difference is important, as some reactions are catalyzed by more than one enzyme, some reactions are not gene-associated (e.g. spontaneous reactions), some enzymes catalyze more than

one reaction, and some genes encode proteins with similar function. However, an automated analysis of all single-gene knockouts would require the accurate representation of gene-reaction associations, which are at present outside the scope of *metano*. Gene-reaction associations are best represented in electronic databases, while *metano* is specifically designed to operate on ASCII files. It is, however, straightforward to implement an *in-silico* analysis of all single-gene knockouts by successively applying the methods of *metano* to scenarios derived from a more comprehensive database representation (cf. [187]).

The automated knockout analysis implemented in *metano* supports three methods for predicting the flux distribution for the knockout scenarios: FBA, MOMA, and wMOMA. While FBA is a less accurate tool for predicting the flux distribution in perturbed networks than MOMA (see Section 3.5), it also has a much lower running time and can give an upper bound on the possible biomass flux: If FBA already predicts a knockout to be lethal, that scenario need not be examined using MOMA or wMOMA.

As the analysis of all scenarios where exactly one reaction is knocked out is a very computationally intensive process, particularly when using MOMA/wMOMA, the analysis has been parallelized under the dispatcher-worker paradigm [188] using the Message-Passing Interface (MPI) [178].

4.6 Metabolite flux minimization

Like *metano*'s implementations of FBA and FVA, the implementation of metabolite flux minimization in *metano* employs iterative dead-end analysis as a preprocessing step in order to speed up the analysis. The flux through any dead-end metabolite or blocked reaction is necessarily zero due to the steady-state condition. Therefore, all dead ends and blocked reactions can be omitted in the analysis, which reduces both the number of optimizations and the dimension of each optimization problem. As described in Section 3.9, MFM requires that flux variables be split into non-negative components.

Since the algorithm of MFM is very similar to FVA, it is also amenable to the 'fastFVA' strategy, where each optimization is started from the previous optimal solution [163]. MFM has been implemented with this speedup in *metano*.

Like FVA, MFM requires the value of the FBA optimum of the objective function. If this value is not supplied, FBA is performed prior to the analysis.

4.7 Model assertion monitor

The algorithm for automated plausibility checking of metabolic models described in Section 3.10 has been implemented in *metano* in the Python class ModelWatcher. This class defines an assertion monitor, which is associated with a model and can then be called repeatedly, e.g. to check for biological plausibility under different growth scenarios.

The assertions are either given as a list of strings or as lines in an ASCII file. The assertions contain symbolic references to reaction fluxes, metabolite fluxes, split ratios, and the minimum and maximum flux values determined by FVA. These identifiers are substituted by actual Python variable references prior to evaluation of the assertions. This substitution is performed via regular expressions [189].

The actual evaluation of assertions is performed by calling Python's built-in `eval()` function. It should be noted that this allows the execution of arbitrary code inserted in the assertion file. It is the user's responsibility to use only Python functions that do not produce disruptive side effects. While this approach has been deemed acceptable for the present application, it makes *metano*'s model assertion monitor unsuitable for execution in a non-safe environment, such as in a web-based application. Syntax checking and actual evaluation of the assertions have been implemented in separate functions, as syntax checks can be performed without actually running any potentially time-consuming numerical analyses on the model.

The following symbolic identifiers are defined: Reaction fluxes are identified by reaction names as they appear in the reaction file. By analogy, the flux through a metabolite node is identified by the corresponding metabolite name. Split ratios are represented by the strings `inratio(<metab>, <rea>)` and `outratio(<metab>, <rea>)` respectively, where <metab> is a metabolite name and <rea> is a reaction name. The two symbols represent a producing ('inratio') or consuming ('outratio') flux through metabolite node <metab> via reaction <rea>. Similarly, the maximum and minimum values for the flux through reaction <rea> as determined by FVA are addressed as `min(<rea>)` and `max(<rea>)`, respectively.

The substitution of the symbolic references involves a search for non-overlapping longest matches of reaction and metabolite names. For this to work, no reaction may have the same name as any metabolite. Processing longest matches first ensures that the substitution algorithm behaves correctly for identifiers that are substrings of other identifiers. For instance, the model iJO1366 contains both the metabolite 'AMP' and the reaction 'AMP_nucleosidase'. Processing the shorter string before the longer would lead to the nonsensical concatenation of a variable reference to the flux through AMP and the non-matching string '_nucleosidase'.

4.8 Free-energy-based assignment of reaction directions

The computation of the lower and upper bounds of the changes in Gibbs free energies associated with the reactions in a metabolic network and the actual assignment of reaction directions have been implemented in two separate programs in *metano*.

The first script reads in a metabolic model from a reaction file, a list of minimum and maximum physiological concentrations for different metabolites, a list of Gibbs energies of formation, and the temperature in degrees Celsius. The script computes lower and upper bounds for the Gibbs free energy changes according to the algorithm described in Section 3.11. Only those reactions are processed for which Gibbs energies

of formation and concentration ranges are given for each reactant and product. Equation 11 is not applied directly in the form shown in Section 3.11. For reasons of numerical stability, the term $\ln(\prod_i c_i^{s_{ij}})$ is computed as $\sum_i s_{ij} \ln c_i$.

The second script reads a metabolic model from a reaction file and a set of flux bounds from a scenario file and a list of lower and upper bounds of Gibbs free energy changes as computed by the first script. It then creates a new set of flux bounds based on the Gibbs free energy ranges, as described in Section 3.11. The new flux bounds are combined with the original bounds and written to a new scenario file, and all changes and conflicts are reported.

5 Application to published metabolic models

Existing studies on genome-scale metabolic models tend to focus on large-scale analyses of the qualitative model predictions for various experimental scenarios. The most common of these analyses are *in-silico* knockout studies and *in-silico* analyses of the qualitative metabolic behavior depending on media conditions. In an *in-silico* knockout analysis, hundreds to thousands of single- or double-gene knockouts are generated *in silico* and evaluated with respect to a small number of biological measures, such as biomass flux, respiration rate, or export rate of a metabolic product [79,190–193]. The parameters that are varied in studies assessing the qualitative predictions of a metabolic model in different growth scenarios include the carbon source, the nitrogen source, and the rate of oxygen supply [77,78,194–196]. Owing to its efficient implementations (see Sections 4.3 to 4.6), *metano* is well suited for such large-scale studies and has been successfully applied in analyses involving thousands to hundreds of thousands of simulations [187,197].

In contrast, the goal of the present study was to analyze the quantitative metabolic behavior for one specific scenario in detail. Experimental results, particularly from fluxomics experiments and gene deletion studies, were employed for assessing the accuracy of the model's predictions for the scenario under study. Parts of the results described and discussed in the following subsection (Section 5.1) have been published earlier [170].

5.1 Escherichia coli

Flux balance analysis was performed on the model iJO1366, the most recent metabolic reconstruction of *E. coli*, for carbon-limited aerobic growth in a defined high-glucose medium. The studied growth scenarios are described in detail in Section 3.1.1.2. In the FBA solution for the original scenario, the biomass flux is 0.85 h^{-1}, the oxygen uptake rate is 16.5 mmol gDW^{-1} h^{-1}, and the CO_2 emission rate is 18.2 mmol gDW^{-1} h^{-1}. 552 of the 2,583 reactions (21%) involving 557 of the 1,805 metabolites (31%) were predicted to be active (at a cutoff of 10^{-12} mmol gDW^{-1} h^{-1}). Split-ratio analysis on the FBA solution revealed that 401 of the 557 metabolites carrying a flux are each produced via a single reaction and consumed via another single reaction. Thus, these metabolite nodes are purely transitory in the scenario under consideration. Only the remaining 156 metabolites (8.6% of all metabolite nodes or 28% of the active metabolites) are branch points, where fluxes split or join.

5.1.1 Balance sheets of currency metabolites

Inside the cell, chemical energy is transferred primarily in two forms: energy-rich phosphate bonds and strong reducing agents. Energy-rich phosphates that act as cellular

energy carriers include nucleoside triphosphates (most importantly ATP) and phosphoenolpyruvate (PEP) [55]. Cellular reducing agents include the free nicotinamide dinucleotides NADH and NADPH, the protein-bound flavin nucleotides $FADH_2$ and $FMNH_2$, membrane-associated isoprenoid quinols such as ubiquinol and menaquinol, the tripeptide glutathione, and redox proteins such as ferredoxins, thioredoxins, and cytochromes [198]. Most of these reducing agents are either proteins, protein-bound, or membrane-associated, and many act only in specific pathways such as the detoxification of free radicals. Only NADH and NADPH act as flexible 'currency metabolites'.

In order to characterize the energy metabolism as predicted by the model, the balance sheets of all relevant currency metabolites were compiled using split-ratio analysis on the FBA solution for the original scenario. The balance sheets (Tables 2 to 9) list for each reaction either the pathway to which it belongs or the biomass components produced from the reaction's main products. As this information was compiled using branch point analysis, it reflects the actual metabolic fate of the reaction products as predicted by FBA for the scenario under consideration.

As only two D-amino acids are active in the FBA solution (D-alanine and D-glutamate), each of which is produced from the respective L-amino acid by a racemase, unmarked names of amino acids will hereafter be used to denote the L-forms.

ATP

ATP is produced (and consumed) at a rate of 69.7 mmol gDW^{-1} h^{-1}. The ATP-producing reactions are shown in Table 2 along with their relative contributions. More than three-fifths of the total ATP (61.4%) is produced in oxidative phosphorylation. The other ATP-producing processes are glycolysis (25.5%), the conversion of acetyl-CoA to acetate (8.5%), recycling of diphosphate (4.0%), and the TCA cycle (0.7%).

Table 2: Split ratios of ATP-producing fluxes (E. coli). Total flux: 69.7 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Reaction/Enzyme	**Pathway/Product**	**Ratio**
ATP synthase	oxidative phosphorylation	61.35%
phosphoglycerate kinase	glycolysis	25.47%
acetate kinase	acetate export	8.46%
polyphosphate kinase	recycling of diphosphate	4.02%
succinyl-CoA synthetase	TCA cycle	0.71%

This result conforms to the expectation that under aerobic growth conditions, the vast majority of ATP is produced via oxidative phosphorylation [199].

Table 3: Split ratios of ATP-consuming fluxes (E. coli). Total flux: 69.7 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown. Abbreviations: AICAR – 5-aminoimidazole-4-carboxamide ribotide, PRPP – phosphoribosyl pyrophosphate

Reaction/Enzyme	**Pathway/Product**	**Ratio**
Biomass (incl. GAM)		66.39%
phosphofructokinase	glycolysis	8.77%
ATP maintenance requirement	NGAM	4.52%
adenylate kinase	AMP→ADP	2.73%
acetyl-CoA carboxylase	fatty acids	2.68%
AICAR biosynthesis from PRPP (5 reactions)	purines	2.67%
glutamine synthetase	glutamine	2.13%
aspartate kinase	threonine, methionine, lysine, peptidoglycans	1.28%
PRPP synthetase	pentose phosphate pathway	1.11%
nucleoside-diphosphate kinase	GTP	0.92%
carbamate kinase	pyrimidines, arginine, polyamines	0.75%
homoserine kinase	threonine	0.65%
nucleoside-diphosphate kinase	UTP	0.49%
shikimate kinase	shikimate pathway	0.46%
UMP kinase	pyrimidines	0.44%
acetylglutamate kinase	arginine, polyamines	0.40%
argininosuccinate synthase	arginine	0.35%
adenylyl-sulfate kinase	sulfur metabolism	0.30%
sulfate adenyltransferase	sulfur metabolism	0.30%
guanylate kinase	purines	0.29%
GMP synthase	purines	0.29%
asparagine synthetase	asparagine	0.29%
glutamate 5-kinase	proline	0.26%
cytidylate kinase	pyrimidines	0.20%
CTP synthase (glutamine)	pyrimidines	0.19%
glucose-1-phosphate adenylyltransferase	glycogen	0.19%
nucleoside-diphosphate kinase	CTP	0.17%
6 reactions in cell wall synthesis	peptidoglycans	0.17%
7 reactions in lipopolysaccharide biosynthesis	lipopolysaccharide	0.13%
phospholipid export (2×3 reactions)	phospholipids	0.11%
ATP phosphoribosyltransferase	histidine, purines	0.11%

Table 3 lists the split ratios of the reactions that consume ATP in the scenario under study. Two-thirds of the ATP (66.2%) is expended as growth-associated maintenance energy (GAM) in the virtual biomass reaction. Glycolysis also consumes a substantial

fraction of the total ATP (8.8%). In addition, 4.5% of the ATP is consumed in non-growth-associated maintenance processes (NGAM), and 2.7% is used for the (re)phosphorylation of AMP to ADP. The major part of the remaining 17.6% is utilized in anabolic pathways, including amino acid biosynthesis (6.2%), purine metabolism (3.4%), fatty acid biosynthesis (2.7%), and pyrimidine metabolism (1.2%). 1.6% of the ATP is converted to the other nucleoside triphosphates GTP (0.9%), UTP (0.5%), and CTP (0.2%). The synthesis of phosphoribosyl pyrophosphate in the pentose phosphate pathway consumes 1.1% of the ATP, while 0.6% is used for the assimilation of sulfur, which is taken up by the cell in the form of sulfate. 0.2% of the ATP enters the biomass directly (via the biomass reaction), and small fractions are consumed in further anabolic pathways, including those producing peptidoglycans, glycogen, deoxyribonucleoside triphosphates, lipopolysaccharide, phospholipids, and various cofactors.

Within the amino acid biosynthetic pathways, glutamine biosynthesis accounts for the largest fraction of the ATP consumption (2.1% of the total ATP flux), followed by threonine (1.3%) and arginine (1.1%). ATP is further needed in the biosyntheses of lysine, asparagine, proline, methionine, histidine, and the aromatic amino acids phenylalanine, tyrosine, and tryptophan. In addition, cysteine biosynthesis is dependent on hydrogen sulfide, which is produced from sulfate via the ATP-dependent sulfur assimilation pathway.

The diphosphate recycling reaction (ADP + diphosphate ⇌ ATP + phosphate) regenerates some ‘spent’ ATP. When considering the net ATP balance, it is therefore necessary to subtract the flux through this reaction from both sides. In addition, glycolysis both consumes and produces ATP. The lower of the two fluxes, i.e. the ATP-consuming flux of 6.1 mmol gDW^{-1} h^{-1} should be subtracted as well. The remaining, non-cyclic flux through ATP is 60.8 mmol gDW^{-1} h^{-1}, i.e. the predicted net ATP production rate is 5.5 moles per mole of glucose.

GTP, UTP, CTP

The total fluxes through the other nucleoside triphosphates are low (GTP: 0.64, UTP: 0.34, CTP: 0.25 mmol gDW^{-1} h^{-1}). Using metabolite flux minimization, these fluxes were found to be essential for biomass production – in fact, all three values correspond to the minimal values necessary for optimal biomass production. However, while the fluxes through GTP, UTP, and CTP are essential, they are not contributing substantially to energy metabolism, which is dominated by ATP and phosphoenolpyruvate.

GTP, UTP, and 47.1% of the CTP are produced from ATP and the corresponding diphosphate via nucleoside-diphosphate kinase, while 52.9% of the CTP is produced from UTP via CTP synthase. The reactions consuming GTP, UTP, and CTP are shown in Tables 4 to 6 with their respective contributions.

Table 4: Split ratios of GTP-consuming fluxes (E. coli). Total flux: 0.64 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Reaction/Enzyme	Pathway/Product	Ratio
adenylosuccinate synthase	purines	38.76%
sulfate adenyltransferase	sulfur metabolism	33.08%
Biomass		27.91%
GTP cyclohydrolase I	folate biosynthesis	0.12%
GTP cyclohydrolase II	flavin biosynthesis	0.06%

Table 5: Split ratios of UTP-consuming fluxes (E. coli). Total flux: 0.34 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Reaction/Enzyme	Pathway/Product	Ratio
CTP synthase	CTP	38.90%
Biomass		35.06%
UDP-N-acetylglucosamine diphosphorylase	peptidoglycans, lipopolysaccharide	17.88%
UTP-glucose-1-phosphate uridylyltransferase	lipopolysaccharide	8.16%

Table 6: Split ratios of CTP-consuming fluxes (E. coli). Total flux: 0.25 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Reaction/Enzyme	Pathway/Product	Ratio
CDP-diacylglycerol synthetase (3 reactions)	phospholipids	45.01%
Biomass		44.21%
3-deoxy-manno-octulosonate cytidylyltransferase	lipopolysaccharide	8.33%
2-C-methyl-D-erythritol 4-phosphate cytidylyltransferase	isoprenoid biosynthesis	2.26%
phosphopantothenate-cysteine ligase	coenzyme A	0.20%

38.8% of the GTP is consumed in the *de-novo* biosynthesis of AMP, 33.1% is used in sulfur assimilation, while 27.9% enters the biomass. GTP is also the starting molecule in folate (0.12%) and flavin biosynthesis (0.06%). 38.9% of the total UTP is converted to CTP, while 35.1% enters the biomass, and 13.8% is consumed in peptidoglycan biosynthesis. Another 12.2% is utilized for the biosynthesis of lipopolysaccharide. The main processes consuming CTP are phospholipid biosynthesis (45.0%), biomass formation (44.2%), and lipopolysaccharide biosynthesis (8.3%). Small amounts of CTP are also needed in isoprenoid (2.2%) and coenzyme A biosynthesis (0.2%).

Phosphoenolpyruvate (PEP)

The flux through PEP is 16.3 mmol gDW^{-1} h^{-1}. As shown in Table 7, all PEP is produced via a single reaction in glycolysis but consumed in a number of reactions in different pathways. FVA was used to verify that for the biomass flux to be optimal, PEP

cannot be produced via any other way than the enolase reaction. This is expected for growth on glucose, as PEP production via other reactions (PEP carboxykinase and PEP synthase, respectively) occurs mainly in gluconeogenesis and is associated with ATP hydrolysis [200,201].

Table 7: Phosphoenolpyruvate (PEP) balance (E. coli). Total flux: 16.3 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
enolase	glycolysis	100%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glucose import (phosphotransferase system)	glycolysis	67.49%
PEP carboxylase	anaplerotic reactions	15.35%
dihydroxyacetone kinase	glycolysis	12.99%
3-phosphoshikimate 1-carboxyvinyltransferase	shikimate pathway	1.95%
3-deoxy-7-phosphoheptulonate synthase	shikimate pathway	1.95%
UDP-N-acetylglucosamine 1-carboxyvinyltransferase	peptidoglycans	0.14%
3-deoxy-8-phosphooctulonate synthase	lipopolysaccharide	0.13%

E. coli uses a phosphotransferase system for the uptake and phosphorylation of mono- and oligosaccharides. In the process, PEP is converted to pyruvate. In the FBA solution for the original scenario, the phosphotransferase system accounts for two-thirds of the total PEP flux (67.5%). Anaplerotic replenishing of the TCA cycle via PEP carboxylase accounts for the second largest fraction (15.4%). 13.0% of the PEP is consumed in the phosphorylation of dihydroxyacetone to dihydroxyacetone phosphate. In the shikimate pathway, PEP is consumed in two reactions. Together they consume 3.9% of the total PEP. In addition, smaller fractions of PEP are consumed in the biosynthesis of peptidoglycans and lipopolysaccharide.

As the phosphotransferase system is the main mode of glucose uptake in *E. coli* [202], the large PEP flux through the phosphotransferase system is expected for a scenario with rapid glucose uptake.

In contrast, the high flux through dihydroxyacetone (DHA) kinase predicted by FBA is unexpected. The flux through DHA is 2.1 mmol gDW^{-1} h^{-1}, 99.7% of which is converted to dihydroxyacetone phosphate by DHA kinase. DHA, while being a usable carbon source [203], is not known as an intracellular metabolite in *E. coli* except in the degradation of glycerol [204,205]. In the FBA solution, glycerol degradation accounts for only 0.26% of the DHA flux. FVA confirms that the flux through glycerol dehydrogenase has a tight range of 4–19 µmol gDW^{-1} h^{-1} under the constraint of optimal biomass production. 100% of the DHA is produced from the cleavage of fructose 6-phosphate (F6P) to glyceraldehyde 3-phosphate and DHA by the enzyme F6P

aldolase. While the F6P aldolase reaction is active in the FBA solution, FVA revealed that it need not be active for optimal growth. In addition, the minimum flux through DHA required for optimal growth is 5.5 μmol gDW^{-1} h^{-1}, as computed via MFM.

F6P aldolase was described in *E. coli* in 2000 [206]. However, nothing is known about its function or expression *in vivo*. Glycolysis in *E. coli* has been studied for more than seven decades and is known to proceed via the Embden-Meyerhof-Parnas pathway [207]. It involves the phosphorylation of F6P to fructose 1,6-bisphosphate via phosphofructokinase and the cleavage of fructose 1,6-bisphosphate to glyceraldehyde 3-phosphate and dihydroxyacetone phosphate via fructose-bisphosphate aldolase [200]. In the model, F6P aldolase is an alternative to fructose-bisphosphate aldolase because the reaction pair F6P aldolase + DHA kinase is stoichiometrically equivalent to the pair phosphofructokinase + fructose-bisphosphate aldolase except for using a different phosphorylating agent (PEP instead of ATP). As noted above, this alternative pathway has, however, never been observed *in vivo*. Moreover, the discoverers of F6P aldolase note that the catalyzed reaction is 10 kJ mol^{-1} more endergonic than the reaction catalyzed by fructose-bisphosphate aldolase [206]. The reaction is therefore unlikely to proceed in the cleavage direction under the conditions found in the cell.

Therefore, the model's predictions would conform more closely to experimental observations if the F6P aldolase reaction were removed or at least constrained to the condensation direction. An FBA with cleavage of F6P disallowed (via an upper bound of zero on the F6P aldolase flux) yielded a flux distribution where 99.3% of all F6P is converted to fructose 1,6-bisphosphate via phosphofructokinase. The flux distribution in this case corresponds to the Embden-Meyerhof-Parnas pathway as it is described for *E. coli*.

Net balance of glycolysis

As glycolysis in *E. coli* produces both ATP and PEP as energy-rich phosphates, a net balance can be compiled by combining the balance sheets for these two metabolites. In the FBA solution for the original scenario, glycolysis (including the phosphotransferase system) generates 1.6 molecules of ATP and 1.5 PEP per molecule of glucose and consumes 0.6 ATP and 1.2 PEP. Hence, the net balance is positive with an output of 1.1 ATP and 0.3 PEP per glucose molecule (all values rounded to two significant digits). In the FBA solution, glycolysis thus operates at 67.4% of its theoretical maximum yield of 2 molecules of ATP or PEP per molecule of glucose [55], while one-third of the metabolic flux (32.6%) branches from the superpathway of glycolysis and the pentose phosphate pathway towards anabolic pathways.

NADH

In the FBA solution for the original scenario, the rate of NADH production and consumption is 33.7 mmol gDW^{-1} h^{-1}. The split ratios are shown in Table 8. Nearly 85% of the total NADH is produced in glycolysis (52.7%) and the subsequent oxidative decarboxylation of pyruvate to acetyl-CoA (32.1%). Another 8.1% is generated in the

TCA cycle. The remaining NADH arises as a byproduct of various amino acid biosyntheses (6.2%), purine biosynthesis (0.6%), and folate metabolism (0.1%).

Table 8: NADH balance (E. coli). Total flux: 33.7 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glyceraldehyde-3-phosphate dehydrogenase	glycolysis	52.74%
pyruvate dehydrogenase	glycolysis/TCA cycle	32.14%
malate dehydrogenase	TCA cycle	5.35%
phosphoglycerate dehydrogenase	serine	4.31%
2-oxoglutarate dehydrogenase	TCA cycle	2.78%
3-isopropylmalate dehydrogenase	leucine	1.11%
Inosine monophosphate (IMP) dehydrogenase	purines	0.60%
histidinol dehydrogenase	histidine	0.47%
prephenate dehydrogenase	tyrosine	0.34%
glycine cleavage system	folate metabolism	0.14%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
NADH dehydrogenase	oxidative phosphorylation	94.48%
enoyl-[acyl-carrier-protein] reductase (11 reactions)	fatty acids	5.10%
methylenetetrahydrofolate reductase	methionine	0.40%

NADH is used almost exclusively for the generation of ATP via oxidative phosphorylation (94.5%). Of the pathways accounting for at least 0.05% of the NADH flux, only the biosyntheses of fatty acids (5.1%) and the amino acid methionine (0.4%) use NADH as reducing agent.

These results are in agreement with the general principle that in aerobically growing cells, NADH is chiefly used for ATP generation, while NADPH is used as reducing agent in anabolic pathways [55]. Furthermore, when growing aerobically on carbohydrates, cells are expected to produce NADH primarily via glycolysis and the TCA cycle [55].

NADPH and thioredoxin

The flux through NADPH is 13.9 mmol gDW^{-1} h^{-1}. The reactions that produce or consume NADPH are shown in Table 9 along with their split ratios. While the flux through NADPH is less than half the NADH flux, NADPH is consumed in many more reactions than NADH. In the predicted flux distribution, NADPH stems mainly from the oxidative branch of the pentose phosphate pathway (PPP), which accounts for 80.7% of the NADPH-producing flux. 13.4% is generated in the TCA cycle via the enzyme

isocitrate dehydrogenase, and the remaining 5.9% arises as a byproduct in association with purine metabolism.

Table 9: NADPH balance (E. coli). Total flux: 13.9 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glucose 6-phosphate dehydrogenase	pentose phosphate pathway	40.35%
phosphogluconate dehydrogenase	pentose phosphate pathway	40.35%
isocitrate dehydrogenase	TCA cycle	13.40%
methylenetetrahydrofolate dehydrogenase	purines	5.89%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glutamate dehydrogenase	glutamate	51.65%
3-oxoacyl-[acyl-carrier-protein] reductase (11 reactions)	fatty acids	13.41%
ketol-acid reductoisomerase (2 reactions)	valine, leucine, isoleucine, coenzyme A	6.95%
aspartate semialdehyde dehydrogenase	threonine, methionine, lysine, peptidoglycans	6.43%
sulfite reductase	sulfur metabolism	4.57%
homoserine dehydrogenase	threonine, methionine	4.21%
shikimate dehydrogenase	shikimate pathway	2.28%
dihydrodipicolinate reductase	lysine, peptidoglycans	2.22%
thioredoxin reductase	conversions to other redox carriers	2.14%
N-acetyl-γ-glutamyl-phosphate reductase	arginine, polyamines	2.01%
glutamate 5-semialdehyde dehydrogenase	proline	1.32%
pyrroline 5-carboxylate reductase	proline	1.32%
glycerol 3-phosphate dehydrogenase	phospholipids	1.03%
UDP-N-acetylenolpyruvoylglucosamine reductase	peptidoglycans	0.17%
dihydrofolate reductase	folate metabolism	0.16%
dTDP-4-dehydrorhamnose reductase	lipopolysaccharide	0.05%

In the FBA solution for the original scenario, NADPH is used in 33 reactions with a flux of more than 0.1 μmol gDW^{-1} h^{-1}. More than 90% of the NADPH is consumed in only two groups of pathways – amino acid biosynthesis (77.8%) and fatty acid synthesis (13.4%). The provision of glutamate alone accounts for 51.7% of the total NADPH consumption. The other amino acid biosynthetic pathways with direct use of NADPH are those of threonine (6.5% of the total NADPH), lysine (4.1%), leucine (2.7%), proline (2.6%), valine (2.5%), methionine (1.9%), arginine (1.8%), isoleucine (1.7%), phenylalanine (1.1%), tyrosine (0.8%), and tryptophan (0.3%). The biosyntheses of all three branched-chain amino acids together consume 7.0%, and those of the three

aromatic amino acids combined account for 2.3%. Moreover, NADPH is consumed in sulfur assimilation (4.6%); the provision of reduced thioredoxin (2.1%); the biosynthesis of phospholipids (1.0%), polyamines, peptidoglycans, lipopolysaccharide, and various cofactors; and folate metabolism (0.16%).

The majority of the reduced thioredoxin is likewise used in sulfur assimilation (71.1%), while 28.9% is consumed in the biosynthesis of deoxyribonucleotides. In the FBA solution, thioredoxin reductase (EC 1.6.4.5), which uses NADPH as electron donor, is the only source of reduced thioredoxin. The flux through reduced thioredoxin is 0.30 mmol gDW^{-1} h^{-1}.

The predicted ratios of the NADPH-producing fluxes are in disagreement with *in-vivo* values for *E. coli* grown aerobically on glucose that were determined by ^{13}C flux analysis: While 35–45% of the NADPH stems from the oxidative PPP, another 35–45% is produced by a proton-translocating transhydrogenase, and 20–25% is generated by isocitrate dehydrogenase [208]. The transhydrogenase PntAB (EC 1.6.1.2) is a membrane-integral protein that converts $NADP^+$ and NADH to NADPH and NAD^+ at the cost of proton flux from the periplasm to the cytosol. Activity of this enzyme thus reduces the number of protons available for ATP generation in oxidative phosphorylation. This energy cost, which induces an increase in respiration, is likely the reason why, according to FVA, at optimal biomass flux, no transhydrogenase activity is possible in the model. The experimental results and an FVA for optimal biomass flux agree that there is no significant NADPH production via malic enzyme (EC 1.1.1.40) in the studied scenario [208].

In an exploratory attempt to reconcile the model's predictions with the experimental observations, FBA was performed for an alternative scenario, which differs from the original scenario in the lower bound of the transhydrogenase flux (5 mmol gDW^{-1} h^{-1} instead of zero). The FBA solution for the alternative scenario reproduces experimentally measured values for the growth rate and the rates of CO_2 emission and O_2 uptake similarly well as the FBA solution for the original scenario (see Appendix B). In contrast to the solution for the original scenario, the ratios between NADPH-producing reactions predicted by FBA for the alternative scenario are consistent with the experimental results [208]: 37% is produced in the oxidative PPP, 36% results from transhydrogenase activity, and 21% is produced by isocitrate dehydrogenase.

The observation that NADPH is involved in many more reactions than NADH while having a lower total flux than NADH conforms as well to the paradigm that NADH is used primarily in energy metabolism while NADPH is used as electron donor in redox reactions in anabolic pathways [55].

The predictions for thioredoxin correspond to known metabolic roles of thioredoxin and thioredoxin reductase [209]. However, MFM revealed that in the model, flux through thioredoxin is not essential for optimal aerobic growth on glucose (see also Section 5.1.4). The reason is that the model contains alternatives to the reactions involving thioredoxins. The alternative reactions use other reducing agents, mainly

NADPH and reduced glutathione. Thioredoxins are known to be redundant in *E. coli* growing under steady-state conditions – *trxA trxC* double mutants are phenotypically inconspicuous under non-stress conditions [210]. Thioredoxins may, however, be essential for the oxidative stress response [211]. As the response to physiological stress is an inherently non-steady-state process [212], FBA is not applicable to the study of such conditions.

5.1.2 Selected branch points

As mentioned above, only 156 metabolites (8.6% of the metabolite nodes in the network) are branch points. In this subsection, selected branch points are analyzed in detail. The following two goals were pursued in the analyses presented here: To elucidate the metabolic roles of the selected metabolites and to identify the relative contributions of the biochemical pathways branching or joining in the respective branch points. All these analyses are scenario-specific as they were performed on a specific simulation scenario – the original scenario for model iJO1366, unless otherwise noted.

Table 10: Split ratios of 2-oxoglutarate (E. coli). Total flux: 8.12 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
aspartate transaminase	aspartate	29.99%
isocitrate dehydrogenase	TCA cycle	22.96%
phosphoserine transaminase	serine	17.85%
L-alanine transaminase	alanine	5.78%
leucine transaminase	leucine	4.61%
valine transaminase	valine	4.33%
succinyldiaminopimelate transaminase	lysine, peptidoglycans	3.80%
acetylornithine transaminase	arginine, polyamines	3.45%
isoleucine transaminase	isoleucine	2.97%
phenylalanine transaminase	phenylalanine	1.89%
tyrosine transaminase	tyrosine	1.41%
histidinol-phosphate transaminase	histidine	0.97%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glutamate dehydrogenase	glutamate	88.48%
2-oxoglutarate dehydrogenase	TCA cycle	11.51%

2-Oxoglutarate

The total flux through 2-oxoglutarate is 8.1 mmol gDW^{-1} h^{-1}. The corresponding split ratios are shown in Table 10.

The ratios of the producing fluxes reveal that 77.0% of the total 2-oxoglutarate is produced as a byproduct in various transamination reactions. In each of these reactions, a transaminase (EC class 2.6.1) transfers the amino group of glutamate to an acceptor molecule, yielding 2-oxoglutarate as a byproduct. In this context, 2-oxoglutarate can be thought of as 'spent glutamate'. Only 23.0% is synthesized via the TCA cycle. The transaminase reactions form a cycle with glutamate dehydrogenase (EC 1.4.1.4), which produces glutamate from 2-oxoglutarate and ammonia at the expense of NADPH oxidation. In the scenario under consideration, 88.5% of the total 2-oxoglutarate is consumed by this reaction, and only 11.5% is metabolized in the TCA cycle via 2-oxoglutarate dehydrogenase.

A net balance can be compiled by subtracting the cyclic flux of 6.3 mmol gDW^{-1} h^{-1} from both sides. In this net balance, 100% of the 2-oxoglutarate is produced in the TCA cycle, and 50.1% is converted to succinyl-CoA in the TCA cycle, while 49.9% is used to produce glutamate.

Glutamate

Table 11 lists the ratios of the fluxes producing and consuming glutamate. The split ratios reveal the presence of a second transamination cycle in which the amino-group donor is glutamine, which is converted to glutamate in the process.

85.0% is produced from 2-oxoglutarate, while 15.0% stems from transamination reactions with glutamine as amino-group donor. 74.0% is used as amino-group donor by other transamination reactions, while 17.6% is converted to glutamine.

Table 11: Split ratios of glutamate (E. coli). Total flux: 8.46 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glutamate dehydrogenase	TCA cycle	85.02%
amidophosphoribosyltransferase	purines	4.40%
phosphoribosylformylglycinamidine synthase	purines	4.40%
GMP synthase	purines	2.39%
CTP synthase	CTP	1.57%
imidazole-glycerol-3-phosphate synthase	histidine, purines	0.93%
glutamine-fructose-6-phosphate transaminase	peptidoglycans, lipopolysaccharide	0.72%
anthranilate synthase	tryptophan	0.56%

Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
aspartate transaminase	aspartate	28.82%
glutamine synthetase	glutamine	17.57%
phosphoserine transaminase	serine	17.15%
L-alanine transaminase	alanine	5.55%
leucine transaminase	leucine	4.43%
valine transaminase	valine	4.16%
succinyldiaminopimelate transaminase	lysine, peptidoglycans	3.65%
N-acetylglutamate synthase	arginine, polyamines	3.31%
acetylornithine transaminase	arginine, polyamines	3.31%
isoleucine transaminase	isoleucine	2.85%
Biomass		2.59%
glutamate 5-kinase	proline	2.17%
phenylalanine transaminase	phenylalanine	1.82%
tyrosine transaminase	tyrosine	1.36%
histidinol-phosphate transaminase	histidine	0.93%
glutamate racemase	peptidoglycans	0.28%
glutamyl-tRNA synthetase	cobalamin, heme	0.05%

Again, a net balance can be compiled by subtracting all fluxes in the two transamination cycles. The split ratios of the net flux of 0.93 mmol gDW^{-1} h^{-1} are shown in Table 12. The net production of glutamate results only from glutamate dehydrogenase.

Table 12: Split ratios of glutamate discounting transamination cycles. Total flux: 0.93 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction	**Pathway**	**Ratio**
glutamate dehydrogenase	TCA cycle	100%
Consuming fluxes		
Reaction	**Pathway**	**Ratio**
N-acetylglutamate synthase	arginine, polyamines	30.10%
glutamine synthetase	glutamine	23.51%
Biomass		23.51%
glutamate 5-kinase	proline	19.75%
glutamate racemase	peptidoglycans	2.54%
glutamyl-tRNA synthetase	cobalamin, heme	0.49%
dihydrofolate synthase	folate metabolism	0.08%

The largest part of the non-circulating glutamate is converted to arginine (26.4%). 23.5% is converted to glutamine, and another 23.5% enters the biomass directly. 19.8% is converted to proline, which, like arginine and the non-circulating glutamine, enters the biomass as a precursor for protein synthesis. Smaller parts are used in the biosynthesis of polyamines (3.7%), peptidoglycans (2.5%), and various cofactors.

For metabolites that are less promiscuous than the ones discussed so far, i.e. produced and consumed in a smaller number of reactions, the split ratios can be analyzed graphically with full metabolic context using branch point analysis.

Glucose 6-phosphate (G6P)

Figure 13 shows the extended metabolic context of glucose 6-phosphate (G6P). The figure was created using AMEBA with a maximum depth of two (see Section 3.8). All G6P is produced by a phosphotransferase system, which transports glucose into the cell and simultaneously phosphorylates it using phosphoenolpyruvate (PEP) as phosphate and energy donor. As can be seen in Figure 13, this process accounts for the majority of both the PEP consumption and the pyruvate formation. In effect, the first step of glycolysis, glucose uptake and phosphorylation, and the last step, dephosphorylation of PEP to pyruvate, are relatively strongly coupled in the scenario under consideration via the phosphotransferase system [213].

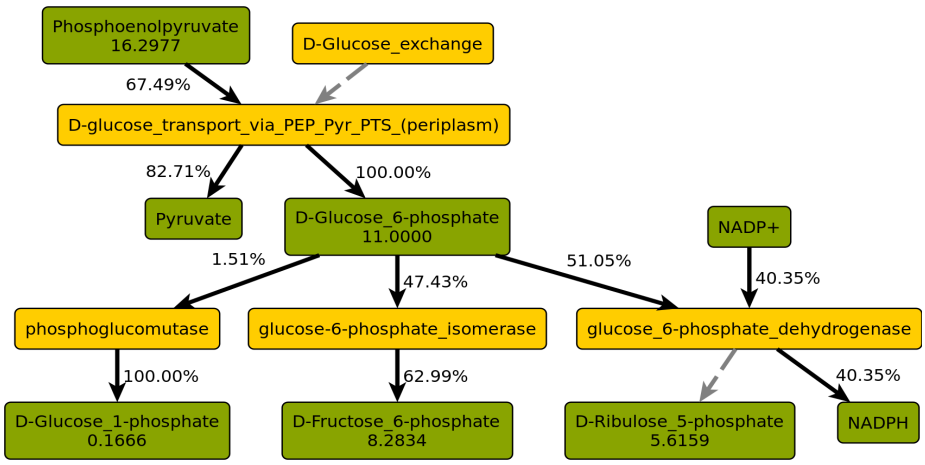

Figure 13: Split ratios of glucose 6-phosphate and adjacent nodes (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

G6P is the first point after glucose uptake where metabolic fluxes split. In the studied scenario, the majority of the G6P (51.1%) enters the pentose phosphate pathway (PPP), while 47.4% is metabolized via glycolysis. 1.5% is converted to glucose 1-phosphate, which is used for the synthesis of glycogen (79.1%) and lipopolysaccharide (20.9%).

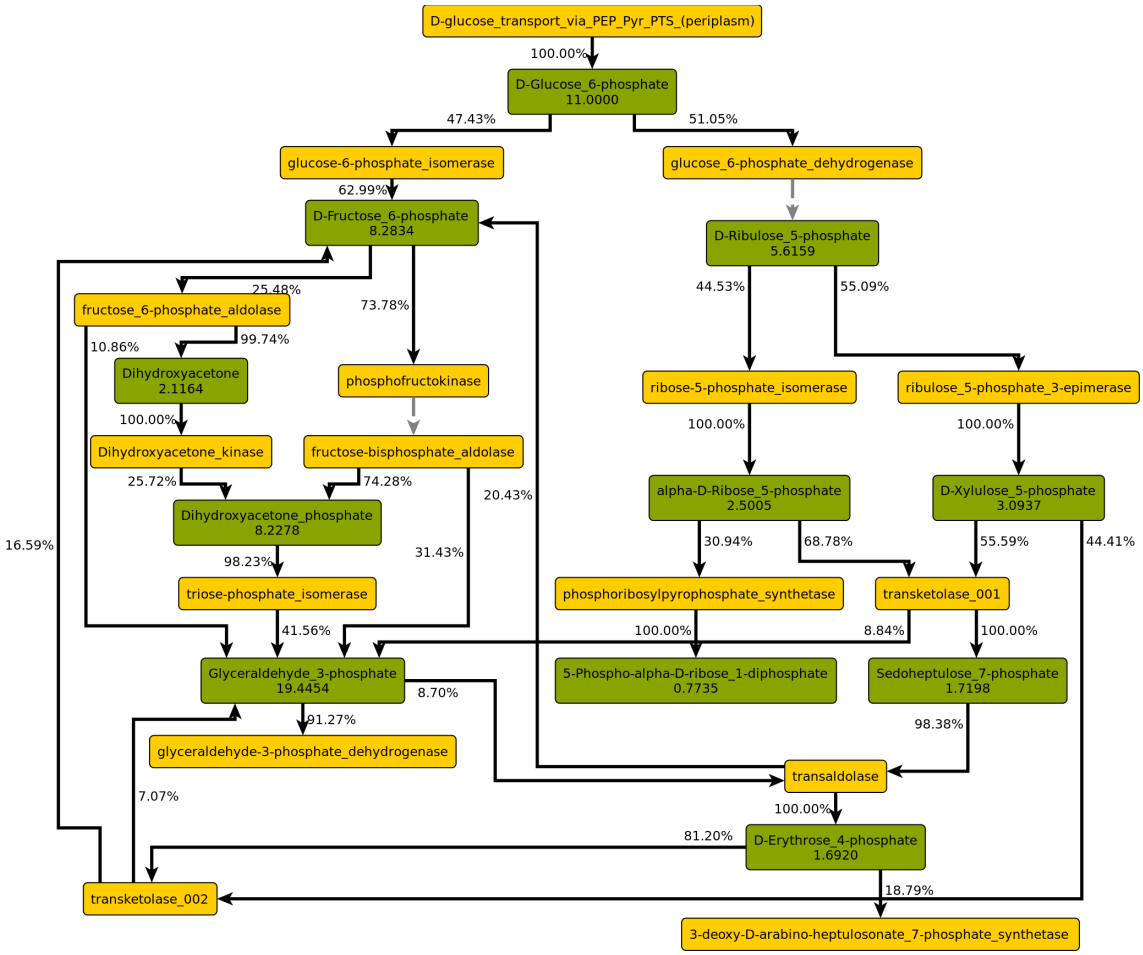

Figure 14: Metabolite fluxes and split ratios within the superpathway of glycolysis and the pentose phosphate pathway (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

The flux through G6P is 11.0 mmol gDW^{-1} h^{-1}, the same as the glucose uptake rate. This corresponds to 66.0 mmol gDW^{-1} h^{-1} of carbon atoms. The flux through PEP, which is produced exclusively in lower glycolysis, is 16.3 mmol gDW^{-1} h^{-1}, which corresponds to a carbon flux of 48.9 mmol gDW^{-1} h^{-1}. This means that 74.1% of the carbon taken up via the phosphotransferase system reaches lower glycolysis. Hence, while more than half of the G6P initially enters the PPP, a large part of the carbon returns to the glycolytic pathway via transketolase and transaldolase. This can be seen in detail in a branch point analysis of the superpathway formed by glycolysis and the PPP, which is shown in Figure 14.

The unexpectedly high flux entering the PPP is explained by the fact that in the model, the enzymes glucose 6-phosphate dehydrogenase (EC 1.1.1.49) and phosphogluconate dehydrogenase (EC 1.1.1.44) of the oxidative part of the PPP constitute the cheapest way to produce NADPH. In the alternative scenario, where a substantial fraction of the NADPH is produced via transhydrogenase (EC 1.6.1.2), only 23.1% of the G6P is predicted to enter the PPP, while 75.4% is converted to fructose 6-phosphate in

glycolysis. In this scenario, the carbon loss via the oxidative PPP is lower than in the original scenario, and 78.8% of the total carbon from glucose reaches PEP (flux through PEP: 17.3 mmol gDW^{-1} h^{-1}, 100% produced via enolase).

Acetate

The fluxes consuming and producing acetate are shown with one additional level of metabolic context in Figure 15. The total flux through acetate is 6.4 mmol gDW^{-1} h^{-1}, which corresponds to the fixed acetate export flux. All of the acetate is exported via acetate/proton symport.

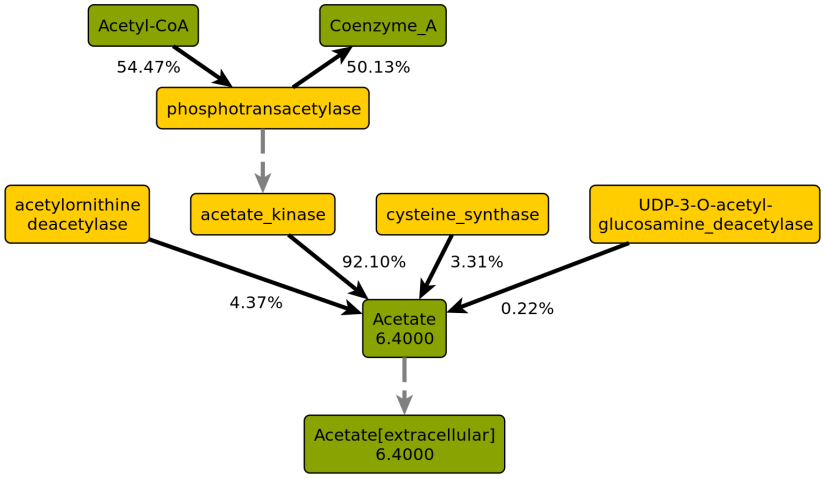

Figure 15: Split ratios of acetate and adjacent nodes (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

While small fractions of acetate arise as a byproduct in arginine metabolism (4.4%), cysteine biosynthesis (3.3%), and the formation of lipopolysaccharide (0.2%), the majority (92.1%) is produced from acetyl-CoA via phosphotransacetylase (EC 2.3.1.8) and acetate kinase (EC 2.7.2.1). In fact, more than half of the total acetyl-CoA (54.5%) is converted to acetate in this way.

If no acetate export flux is enforced, i.e. the corresponding constraint is disabled in the scenario file, the reactions of acetate kinase and phosphotransacetylase are predicted by FVA to proceed in the opposite direction, i.e. recycling the acetate formed as a byproduct to the central carbon metabolism. This means that in the original scenario, acetate is only produced from acetyl-CoA to satisfy the high demand of acetate imposed by forced secretion. According to FVA, this pathway for acetate production is cheapest with respect to the objective of maximizing biomass output.

5.1.3 Metabolic cycles and metabolite pools

As has already been seen in the example of the superpathway of glycolysis and the pentose phosphate pathway shown in Figure 14, branch point analysis (BPA) allows simultaneously analyzing all fluxes in a particular metabolic context. BPA was applied to the study of selected metabolic cycles and metabolite pools.

TCA cycle

The fluxes in the TCA cycle, as predicted by FBA for the scenario under study, are shown in Figure 16. In the original scenario, only about one-sixth of the total acetyl-CoA is metabolized via the TCA cycle. As mentioned above, the largest part of the acetyl-CoA is converted to acetate, which is then secreted.

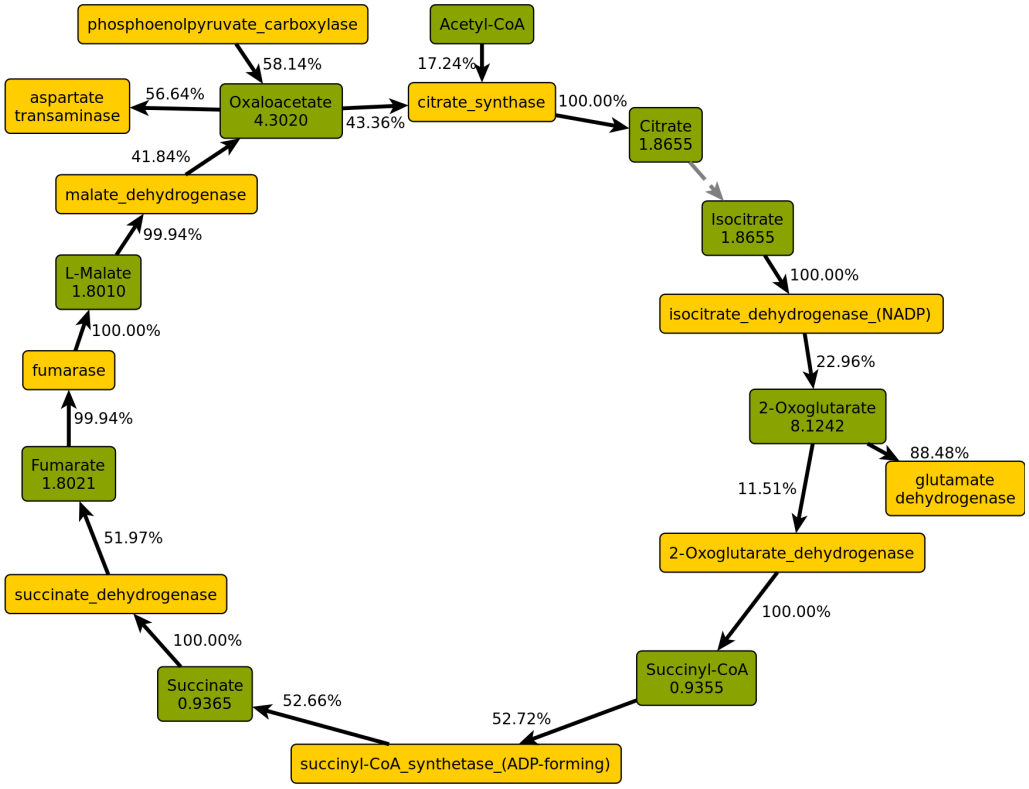

Figure 16: Metabolic fluxes and split ratios in the TCA cycle (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

The path from citrate synthase to 2-oxoglutarate is linear, i.e. all of the citrate produced by citrate synthase is converted to 2-oxoglutarate. As discussed in Section 5.1.2, 2-oxoglutarate is a branch point: While about 50.1% of the 2-oxoglutarate produced in the TCA cycle is metabolized within the TCA cycle, the remaining 49.9% is converted to glutamate, which is the starting molecule for several anabolic pathways.

All of the succinyl-CoA, which is generated from 2-oxoglutarate in the TCA cycle, is further converted to succinate, 52.7% within the TCA cycle, 33.0% in lysine metabolism, and 14.3% in methionine biosynthesis. 100% of the succinate is subsequently converted to fumarate via succinate dehydrogenase, but this reaction accounts for only 52.0% of the fumarate production. Nearly all of the fumarate is converted to malate, virtually all of which is then oxidized to oxaloacetate. In the FBA solution for the original scenario, only 41.8% of the oxaloacetate is produced in the TCA cycle, while the majority (58.1%) is generated anaplerotically from phospho-enolpyruvate.

Oxaloacetate is the second point in the TCA cycle where the metabolic flux splits between the catabolic cycle and anabolic pathways. 56.6% of the oxaloacetate is transaminated to aspartate, which, like glutamate, is the starting molecule in several biosynthetic pathways. However, much of the carbon flux initially directed to aspartate formation is returned to oxaloacetate, as was revealed in a further branch point analysis of the full metabolic context of aspartate and oxaloacetate.

Aspartate transamination cycle

The cycle between aspartate and oxaloacetate identified by BPA, which is shown in Figure 17, consists of four steps, of which two are shared with the TCA cycle: The aspartate generated from oxaloacetate transfers its amino group to different acceptors in arginine and purine metabolism. In each case, fumarate is formed and subsequently converted back to oxaloacetate in two reactions of the TCA cycle.

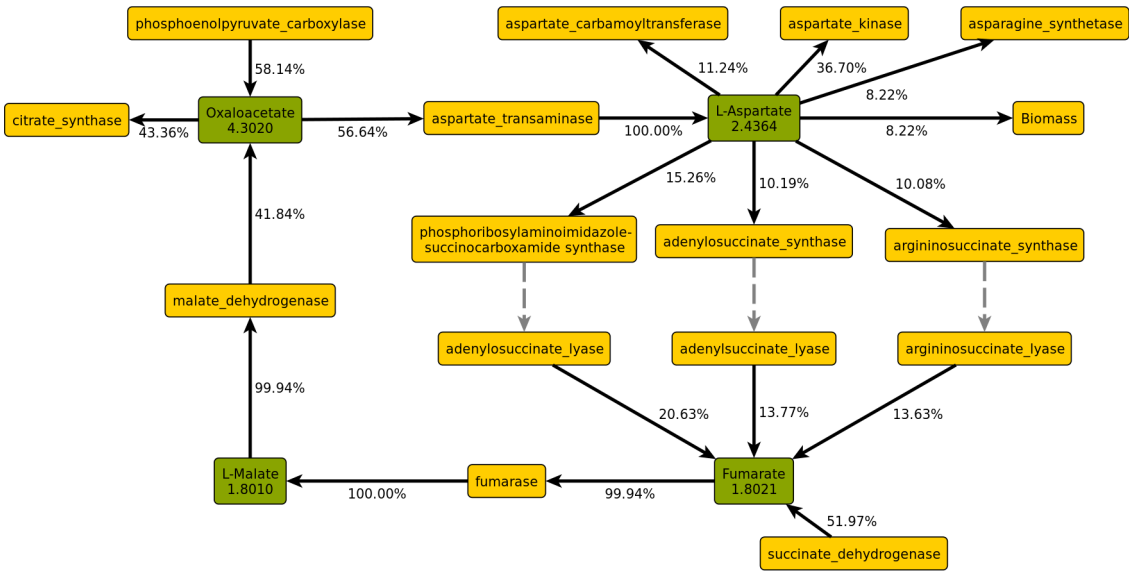

Figure 17: Cycle between aspartate and oxaloacetate (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

This third transamination cycle carries a combined flux of 0.87 mmol gDW^{-1} h^{-1}. Subtracting this flux from both sides of the oxaloacetate balance yields a net balance, in

which 54.3% of the non-circulating oxaloacetate is metabolized in the TCA cycle, while 45.7% is used for the provision of aspartate for anabolic pathways. In the corrected balance, an even larger fraction of the oxaloacetate (72.8%) is produced anaplerotically, while only 27.2% is formed in the TCA cycle.

Comparison of predicted TCA cycle fluxes with experimental data

The prediction that 54% of the acetyl-CoA is converted to acetate for secretion is in fairly good agreement with experimental results: Fischer et al. [154] obtained a range of 55–59% using ^{13}C tracer analysis. In contrast, the experimentally determined values for the fraction of acetyl-CoA entering the TCA cycle (25–30%) and the flux through the TCA cycle (27–33% of the glucose uptake rate) could not be reproduced by FBA for the original scenario. In the predicted flux distribution, only 17% of the acetyl-CoA enters the TCA cycle. Moreover, the flux through isocitrate dehydrogenase is only 17% of the glucose uptake rate.

Considerably larger fluxes in the TCA cycle were obtained when performing FBA for the alternative scenario, where the flux through transhydrogenase is fixed at 5 mmol gDW^{-1} h^{-1}. In this scenario, the predicted flux through isocitrate dehydrogenase is 2.9 mmol gDW^{-1} h^{-1}, which corresponds to 26% of the glucose uptake rate. This value is much closer to the experimentally determined range of 27–33%. As a result of the larger flux through the TCA cycle, the fraction of NADH that is generated in the TCA cycle is also increased (12.9% as opposed to 8.1% in the original scenario). At 24.8% the fraction of acetyl-CoA entering the TCA cycle is also much closer to the experimentally determined range of 25–30% than the value predicted for the original scenario (17.2%).

Tetrahydrofolate pool

Tetrahydrofolate (THFA) is an important carrier of activated one-carbon groups. It can also act as a reducing agent. The fluxes in the folate-dependent one-carbon pool are shown in metabolic context in Figure 18.

The total flux through THFA is 0.98 mmol gDW^{-1} h^{-1}. 99.92% of the THFA circulates, and only 0.08% is produced via *de-novo* synthesis. As the dihydrofolate moiety, once synthesized, is unchanged in all metabolic reactions in the model, *de-novo* synthesis is only active to the extent needed to compensate for the THFA derivatives lost via the biomass reaction. While a small fraction is directly converted to 10-formyl-THFA in the ATP-dependent reassimilation of formate via formate-THFA ligase, nearly 99.8% of the THFA is converted to 5,10-methylene-THFA as the primary activated form. 5,10-methylene-THFA is generated from THFA via enzymatic cleavage of either serine (94.8% of the THFA flux) or glycine (4.9%). 84.0% of the 5,10-methylene-THFA is subsequently converted to 10-formyl-THFA, while 13.8% is reduced to 5-methyl-THFA. 10-formyl-THFA is consumed in two reactions in purine metabolism, where it donates a formyl group. Virtually all of the 5-methyl-THFA is consumed in the last step of methionine biosynthesis, where the activated methyl group is transferred to the thiol group of L-homocysteine. The remaining fractions of the 5,10-methylene-THFA are

used for the synthesis of dTMP (2.2%) and coenzyme A (0.05%). In the first step of coenzyme A biosynthesis, 5,10-methylene-THFA donates a methylene group, while in the synthesis of dTMP from dUMP, it transfers both the activated methylene group and two electrons and is thus converted to dihydrofolate. The dihydrofolate is then reduced back to THFA.

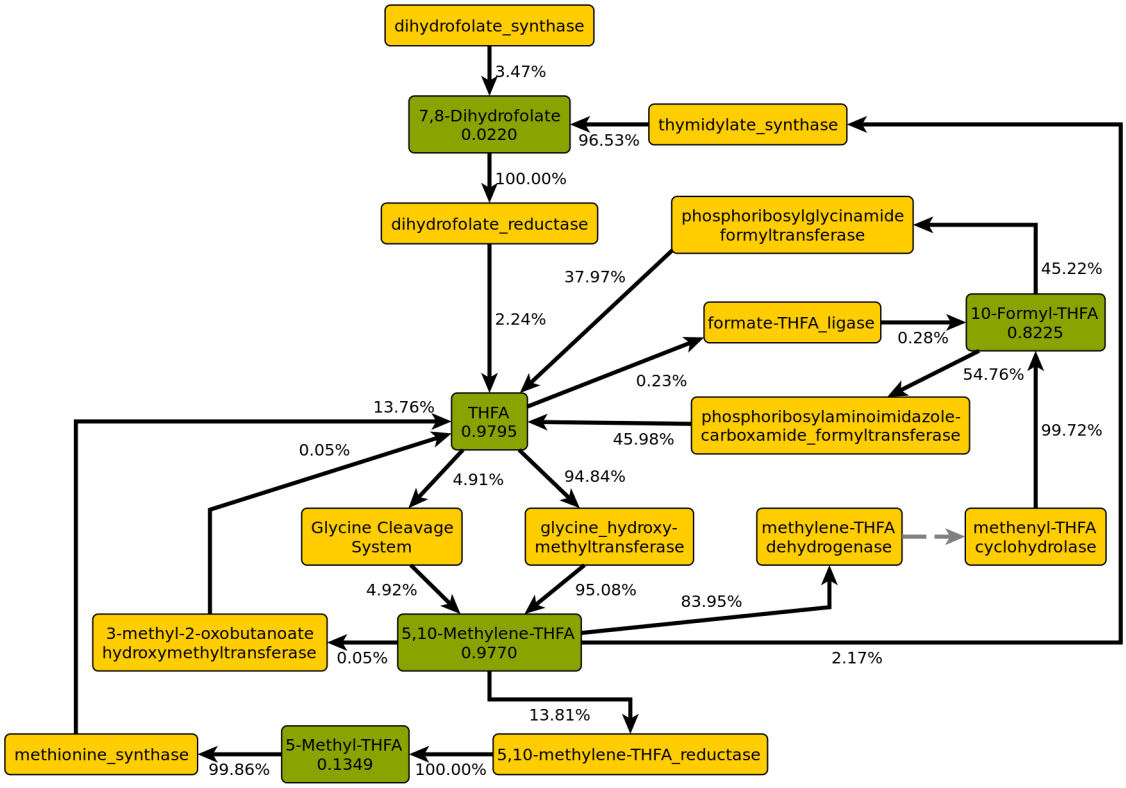

Figure 18: Metabolite fluxes and split ratios within the tetrahydrofolate pool (E. coli). Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

5.1.4 Essentiality of metabolite nodes

Metabolite flux minimization (MFM) was performed for the original scenario to identify metabolites that are essential, i.e. those that must necessarily carry a non-zero flux for growth to be optimal. Of the 557 metabolite nodes carrying a flux, 548 (98%) were identified as essential. Table 13 shows the nine metabolites that are active in the FBA solution but were predicted by MFM to be nonessential. In the model, these metabolites lie on pathways that are not essential for biomass formation in the scenario under consideration. The low number of nonessential metabolites likely results from the large number of constraints present in the studied scenario. Similarly, only 76 of the 552 active reactions were predicted by FVA to be nonessential, while all other active reactions must have a non-zero flux in order for the biomass flux to be optimal.

Table 13: Metabolites carrying a flux but predicted to be nonessential (E. coli). The right column lists the number of reactions in which each metabolite appears in the network.

Metabolite	Number of reactions
adenosine	7
α-D-ribose 1-phosphate	7
fructose 1,6-bisphosphate	4
glycine[periplasm]	4
hydroxypyruvate	3
H+[extracellular]	6
thioredoxin (reduced)	10
thioredoxin (oxidized)	10
[cell wall precursor of two linked disaccharide pentapeptide and tetrapeptide murein units]	3

In the predicted flux distribution for the studied scenario, the active reactions involving adenosine and α-D-ribose 1-phosphate are part of an adenine salvage pathway. The model contains an alternative route, where adenine is converted to AMP in one reaction step instead of two.

In the model, flux through fructose 1,6-bisphosphate is not required for optimal growth because of the sedoheptulose bisphosphate bypass. In this pathway, which was originally described in 2009 for transaldolase-deficient *E. coli* strains, sedoheptulose 7-phosphate is degraded to erythrose 4-phosphate and dihydroxyacetone phosphate by phosphofructokinase and fructose-bisphosphate aldolase [214]. The following reaction sequences allowed by the model are stoichiometrically equivalent:

I (glycolysis, active in FBA solution):

phosphofructokinase : ATP + fructose 6-phosphate → ADP + fructose 1,6-bisphosphate + H+

fructose-bisphosphate_aldolase : fructose 1,6-bisphosphate → dihydroxyacetone phosphate + glyceraldehyde 3-phosphate

II (transaldolase + sedoheptulose bisphosphate bypass):

transaldolase : glyceraldehyde 3-phosphate + sedoheptulose 7-phosphate ← erythrose 4-phosphate + fructose 6-phosphate

phosphofructokinase_(s7p) : ATP + sedoheptulose 7-phosphate → ADP + H^+ + sedoheptulose 1,7-bisphosphate

Sedoheptulose_1,7-bisphosphate_D-glyceraldehyde-3-phosphate-lyase : sedoheptulose 1,7-bisphosphate → dihydroxyacetone phosphate + erythrose 4-phosphate

As noted in Section 5.1.1, glycolysis in *E. coli* has been studied extensively and is known to proceed via the classical Embden-Meyerhof-Parnas pathway. The conversion of fructose 6-phosphate to fructose 1,6-bisphosphate is a central step in this ubiquitous pathway, which is tightly regulated [200]. Therefore, it is highly unlikely that an alternative pathway, which was only discovered in 2009, should play a significant role

in the wild type. Indeed, activity in this pathway has only been observed in transaldolase-deficient mutants [214]. As this pathway uses the glycolytic enzymes phosphofructokinase and fructose-bisphosphate aldolase, whose affinity for the sedoheptulose phosphates is probably dramatically lower than that for the corresponding fructose phosphates, the bypass reactions are unlikely to take place unless there is an accumulation of sedoheptulose 7-phosphate. Hence, the model's predictions when specifically studying the wild type could be made more accurate by disallowing flux through the bypass reactions.

Periplasmic glycine only occurs as an intermediate in the following sequence of transport reactions:

glycine_reversible_transport_via_proton_symport_(periplasm) : glycine[p] + H^+[p] ⟵ glycine + H^+

glycine_transport_in_via_sodium_symport_(periplasm) : glycine[p] + Na^+[p] ⟶ glycine + Na^+

calcium_/_sodium_antiporter_(1_1) : Ca^{2+} + Na^+[p] ⟵ Na^+ + Ca^{2+}[p]

The net effect of this reaction sequence is the simultaneous import of calcium ions and export of protons. While the Na^+/Ca^{2+} antiporter is the only calcium importer available in the model, proton and sodium symporters with the same stoichiometry exist for a variety of organic anions, which can thus replace the first two reactions of the transport sequence. Alternatives include glycolate, acetate, and glutamate.

The two active reactions involving hydroxypyruvate, which are catalyzed by hydroxypyruvate isomerase and hydroxypyruvate reductase (NADH), can be replaced by the single reaction catalyzed by tartronate semialdehyde reductase, in which this intermediate does not occur.

In the FBA solution, excess protons produced by metabolic reactions are first transported to the periplasm via the complexes of the electron transport chain and then secreted to the extracellular space. Alternatively, these protons can, however, also be converted to water in the periplasm via the following reaction, which is catalyzed by a ferroxidase (EC 1.16.3.1):

$$4\ Fe^{2+} + 4\ H^+ + 1\ O_2 \longrightarrow 4\ Fe^{3+} + 2\ H_2O$$

In order to convert all excess protons to water, the rate of Fe^{2+} uptake has to increase by 14.0 mmol $gDW^{-1}\ h^{-1}$, and the Fe^{3+} secretion rate has to increase by the same amount. This extreme rate, which is equivalent to a mass flux of 0.85 grams of iron per gram dry weight per hour, is highly unrealistic, especially as the Fe^{3+} produced in the reaction is barely soluble at pH 7 [215]. Hence, this process should be disallowed in the model by constraining the Fe^{3+} secretion rate to zero (or a value close to zero). The only reason why this possibility to freely import electrons does not have a more dramatic effect on the model's predictions is that the model does not contain any reaction by which the electrons could be transferred to a cytosolic redox carrier.

Thioredoxin occurs in ten reactions in the network, seven of which can have a non-zero flux value according to FVA. However, the model contains stoichiometric alternatives

to all of these reactions, which use different reducing agents. Therefore, thioredoxin is predicted to be nonessential for optimal growth, both in its reduced and in its oxidized form.

Menaquinone-8, which is not active in the examined FBA solution, is the most promiscuous nonessential metabolite. It occurs in twenty-three reactions in the network, none of which needs to be active for optimal growth. FVA identified nine of these reactions as being able to carry a flux.

Table 14: Intracellular metabolites with the largest minimum fluxes (E. coli). Shown are the minimum fluxes required for optimal growth.

Metabolite	**Minimum flux**	**Metabolite**	**Minimum flux**
H^+	239.2	$NADP^+$	13.9
H2O	108.1	NADPH	13.9
ADP	69.7	pyruvate	13.3
ATP	69.7	coenzyme A	11.8
phosphate	66.9	glucose 6-phosphate	11.0
NAD^+	33.7	acetyl-CoA	10.8
NADH	33.7	ammonium	9.4
ubiquinol-8	33.0	L-glutamate	8.5
ubiquinone-8	33.0	dihydroxyacetone phosphate	8.2
CO_2	23.5	2-oxoglutarate	8.1
glyceraldehyde 3-phosphate	17.8	fructose 6-phosphate	6.6
3-phospho-D-glycerate	17.7	acetate	6.4
1,3-bisphospho-D-glycerate	17.7	acetyl phosphate	5.9
O_2	16.5	6-phospho-D-gluconate	5.6
2-phospho-D-glycerate	16.3	6-phospho-D-glucono-1,5-lactone	5.6
phosphoenolpyruvate	16.3	D-ribulose 5-phosphate	5.6

The thirty-two intracellular metabolites with the highest minimum flux sums are shown in Table 14. They include five ubiquitous cofactor or currency metabolites, each in the active and depleted forms (ATP/ADP, NADH/NAD^+, ubiquinol/ubiquinone, NADPH/ $NADP^+$, and acetyl-CoA/coenzyme A), as well as five very small compounds (H^+, H_2O, phosphate, CO_2, and ammonium). This set further contains nine intermediates of glycolysis and three from the pentose phosphate pathway, all from the oxidative part. The glycolytic intermediates have a high minimum flux because all carbon taken up as glucose must initially go through glycolysis, and fluxes split from glycolysis to other pathways only gradually. The large required flux in the first metabolites of the pentose phosphate pathway is explained by the fact that this pathway is the least expensive route in the model for producing NADPH. The remaining five metabolites with high required fluxes are O_2, glutamate, 2-oxoglutarate, acetate, and acetyl phosphate. A high oxygen flux is expected for a rapid aerobic growth scenario, and the large required flux through

glutamate and 2-oxoglutarate is due to the transamination cycle involving these metabolites. The high minimum fluxes through acetate and acetyl phosphate result from the forced acetate secretion.

An alternative view on minimum required metabolite fluxes is presented in Table 15, which shows the twenty largest minimum carbon fluxes for intracellular non-cofactor metabolites. The carbon flux through a metabolite is the product of the metabolite flux and the number of carbon atoms in the compound. In this ranking, glucose 6-phosphate appears at the top instead of the eighth position, as all carbon taken up by the cell has to pass through this intermediate. The list contains another nine metabolites from glycolysis and five intermediates from the pentose phosphate pathway. The other five metabolites with high required carbon fluxes are glutamate, 2-oxoglutarate, CO_2, oxaloacetate, and acetate. Oxaloacetate has a large required flux because it is involved in a transamination cycle, as detailed above.

Table 15: Metabolites with the largest essential carbon fluxes (E. coli). Fluxes are in mmol gDW^{-1} h^{-1}.

Metabolite	**C atoms**	**Minimum flux**	**Minimum C flux**
glucose 6-phosphate	6	11	66
glyceraldehyde 3-phosphate	3	17.75	53.26
3-phospho-D-glycerate	3	17.75	53.24
1,3-bisphospho-D-glycerate	3	17.75	53.24
2-phospho-D-glycerate	3	16.3	48.89
phosphoenolpyruvate	3	16.3	48.89
L-glutamate	5	8.46	42.28
2-oxoglutarate	5	8.12	40.62
pyruvate	3	13.3	39.9
fructose 6-phosphate	6	6.59	39.55
6-phospho-D-gluconate	6	5.62	33.7
6-phospho-D-glucono-1,5-lactone	6	5.62	33.7
D-ribulose 5-phosphate	5	5.62	28.08
dihydroxyacetone phosphate	3	8.23	24.68
CO_2	1	23.5	23.5
acetyl-CoA	2*	10.82	21.64
oxaloacetate	4	4.3	17.21
D-xylulose 5-phosphate	5	3.09	15.47
acetate	2	6.4	12.8
D-ribose 5-phosphate	5	2.5	12.5

* discounting coenzyme A moiety

5.2 Thermotoga maritima

For *T. maritima*, FBA was performed on the scenario defined by the set of upper and lower bounds given in the SBML file of the model (see Section 3.1.2 for details). This scenario describes anaerobic growth on glucose as the sole carbon source. At a glucose uptake rate of 10.0 mmol gDW^{-1} h^{-1}, the predicted growth rate is 0.36 h^{-1}, and the model exports CO_2 at a rate of 16.5 mmol gDW^{-1} h^{-1}, acetate at a rate of 15.4 mmol gDW^{-1} h^{-1}, and hydrogen (H_2) at a rate of 33.2 mmol gDW^{-1} h^{-1}. This means that 51% of the carbon taken up as glucose is secreted as acetate, and another 27% is converted to CO_2. Only 21% of the carbon is incorporated into biomass. In the FBA solution for the studied scenario, 273 of the 645 reactions (42%) have a flux with an absolute value above 10^{-12} mmol gDW^{-1} h^{-1}. The subnetwork spanned by these active reactions contains 276 of the 564 metabolites (49%). Of these 276 metabolites, only 86 (15.2% of all metabolite nodes or 31% of the active metabolites) are branch points. The remaining 190 metabolite nodes lie on unbranched paths.

In accordance with the differences in ecology between the two organisms, the *T. maritima* model exhibits qualitatively different behavior compared to the *E. coli* models: As *T. maritima* is an obligate anaerobe, there is no oxygen uptake. In addition, *T. maritima* does not possess a phosphotransferase system [216,217]. Consequently, 98.4% of the phosphoenolpyruvate is converted to pyruvate via pyruvate kinase (EC 2.7.1.40). In the model, the only usable form of sulfur is elemental sulfur. Therefore, the much more energy-intensive activation and reduction of sulfate is not observed in the FBA solution. In fact, the sulfate reduction pathway is absent from the model and the underlying genome annotation.

T. maritima generates energy by fermentation. In the FBA solution, ATP synthase and all enzymes associated with the electron transfer chain are inactive. Excess intracellular protons are removed via an acetate/proton symporter (80.8%), which accounts for 100% of the acetate export, and a proton-exporting ATPase (19.2%), which hydrolyzes one molecule of ATP per proton. The TCA cycle operates as a branched pathway in the studied flux distribution: The forward branch converts citrate to glutamate, while the reverse pathway transforms oxaloacetate to aspartate via fumarate.

5.2.1 ATP balance

The split ratios of the ATP-producing fluxes are shown in Table 16. The total flux through ATP is 52.5 mmol gDW^{-1} h^{-1}. 71.0% is produced in glycolysis, and 29.0% stems from acetate fermentation. These results are in agreement with the general observation that in the fermentative metabolism of carbohydrates, ATP is produced primarily via substrate-level phosphorylation [200,218].

Table 16: Split ratios of ATP-producing fluxes (T. maritima). Total flux: 52.5 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Reaction/Enzyme	**Pathway/Product**	**Ratio**
phosphoglycerate kinase	glycolysis	35.79%
pyruvate kinase	glycolysis	35.21%
acetate kinase	acetate fermentation	28.97%

The ATP-consuming reactions and their flux split ratios are shown in Table 17. The growth-associated maintenance requirement (GAM) accounts for only 31.2% of the ATP-consuming flux (*E. coli* iJO1366: 66.2%), while non-growth-associated energy-dependent processes are not represented in the *T. maritima* model iTZ478. The largest fraction of the ATP is consumed in the processes of glucose uptake and glycolysis, which together account for 53.1% of the total ATP-consuming flux. Glucose uptake is more expensive than in *E. coli*, as *T. maritima* lacks a phosphotransferase system (PTS) and employs primary active transport via an ABC transporter instead. While the PTS simultaneously imports and phosphorylates glucose using phosphoenolpyruvate as the energy-rich phosphate donor, the ABC transporter hydrolyzes one molecule of ATP just to transport one molecule of glucose into the cell. In the cytosol, glucose is then phosphorylated via a hexokinase at the expense of a second molecule of ATP.

3.5% of the ATP is consumed by a proton-translocating ATPase, which is used to export excess intracellular protons to the medium. Like in *E. coli*, notable fractions of the ATP are consumed in the processes of amino acid biosynthesis (3.6%), the provision of purine and pyrimidine nucleotides (2.2% and 0.8%, respectively), and the (re)phosphorylation of AMP to ADP (2.1%). In the FBA solution, 1.5% of the ATP is used for the synthesis of oxaloacetate from pyruvate via pyruvate carboxylase (EC 6.4.1.1). Furthermore, the provision of phosphoribosyl pyrophosphate in the pentose phosphate pathway accounts for 1.1% of the ATP consumption.

Smaller amounts of ATP are converted to dGTP (0.5%) and UTP (0.2%). Fatty acid biosynthesis accounts for only 0.1% of the ATP consumption (*E. coli* iJO1366: 2.7%). Another 0.1% of the ATP enters the biomass directly. Further small amounts of ATP are consumed in folate metabolism and the biosyntheses of glycogen, peptidoglycans, polyamines, isoprenoid precursors, and various cofactors. In contrast to the *E. coli* flux distribution analyzed in the previous section, no ATP is consumed in sulfur metabolism, as the simulation scenario for *T. maritima* includes elemental sulfur as the sole source of sulfur (*E. coli*: sulfate).

As in the *E. coli* flux distribution, the fluxes through the other nucleoside triphosphates GTP, UTP, and CTP are low (0.20, 0.11, and 0.05 mmol gDW^{-1} h^{-1}, respectively), and they do not strongly contribute to energy metabolism. In the studied scenario, it is not necessary to explicitly examine phosphoenolpyruvate as an energy metabolite, as 98.4% is converted to ATP via pyruvate kinase. The remainder is consumed in metabolic reactions where its primary role is not that of an energy donor.

Table 17: Split ratios of ATP-consuming fluxes (T. maritima). Total flux: 52.5 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown. Abbreviations: AICAR – 5-aminoimidazole-4-carboxamide ribotide, PRPP – phosphoribosyl pyrophosphate

Reaction/Enzyme	**Pathway/Product**	**Ratio**
Biomass (incl. GAM)		31.29%
D-glucose transport via ABC system	glucose import	19.03%
hexokinase	glycolysis	19.02%
phosphofructokinase	glycolysis	15.07%
H^+-exporting ATPase	export of excess protons	3.48%
adenylate kinase	AMP→ADP	2.11%
glutamine synthetase	glutamine	1.65%
pyruvate carboxylase	anaplerotic reactions	1.53%
AICAR biosynthesis from PRPP (4 reactions)	purines	1.33%
PRPP synthetase	pentose phosphate pathway	1.11%
carbamoyl-phosphate synthase (glutamine-hydrolyzing)	pyrimidines, arginine	0.81%
guanylate kinase	purines	0.65%
nucleoside-diphosphate kinase	dGTP	0.48%
shikimate kinase	shikimate pathway	0.28%
aspartate kinase	lysine	0.24%
uridylate kinase	pyrimidines	0.23%
acetylglutamate kinase	arginine, polyamines	0.22%
nucleoside-diphosphate kinase	UTP	0.22%
argininosuccinate synthase, reversible	arginine	0.19%
GMP synthase	purines	0.19%
glutamate 5-kinase	proline	0.14%
acetyl-CoA carboxylase	fatty acids	0.13%
glucose-1-phosphate adenylyltransferase	glycogen	0.11%
3 reactions in folate metabolism	folate metabolism	0.10%
5 reactions in cell wall biosynthesis	peptidoglycans	0.09%
CTP synthase (glutamine)	pyrimidines	0.09%
ATP phosphoribosyltransferase reversible	histidine, purines	0.06%

5.2.2 Balance sheets of redox carriers

Fermentation differs from respiration in the terminal acceptor of the electrons generated in the oxidative breakdown of a reduced carbon source. While respiration utilizes an exogenous electron acceptor such as molecular oxygen, nitrate, or sulfate, the electrons are transferred to an endogenous acceptor in fermentation [55]. In the fermentative catabolism of carbohydrates, the terminal electron acceptor is usually derived from

pyruvate [200]. This difference should be reflected in the balance sheets of the redox carriers. The balance sheets for NADH and NADPH are shown in Tables 18 and 19.

Table 18: NADH balance (T. maritima). Total flux: 19.3 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown. Abbreviations: IMP – inosine monophosphate

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
glyceraldehyde-3-phosphate dehydrogenase	glycolysis	97.52%
3-isopropylmalate dehydrogenase	leucine	0.80%
dihydroorotic acid dehydrogenase	pyrimidines	0.58%
IMP dehydrogenase	purines	0.52%
histidinol dehydrogenase	histidine	0.34%
prephenate dehydrogenase	tyrosine	0.24%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
NAD-linked hydrogenase	hydrogen formation	89.12%
glutamate dehydrogenase (NAD)	glutamate	6.13%
malate dehydrogenase	aspartate (reverse TCA cycle)	3.17%
dihydrodipicolinate reductase (NADH)	lysine	0.66%
glycine cleavage system	glycine	0.46%
methylenetetrahydrofolate reductase	methionine	0.38%

In the FBA solution, the total flux through NADH is 19.3 mmol gDW^{-1} h^{-1}. 97.5% of the NADH is produced in glycolysis, while the remaining NADH arises as a byproduct in various anabolic pathways. 89.1% is used to reduce intracellular protons to molecular hydrogen, which is exported. The remaining NADH is consumed in the biosyntheses of the amino acids glutamate (6.1%), aspartate (3.2%), lysine (0.7%), glycine (0.5%), and methionine (0.4%).

As expected, NADH is primarily formed in glycolysis. The NADH then has to be reoxidized to NAD^+. In the model, this reoxidation occurs primarily via the following hydrogenase reaction:

$$NADH + H^+ \rightarrow NAD^+ + H_2.$$

Intracellular protons thus serve as an endogenous electron acceptor in the scenario under consideration. This particular reaction is, however, thermodynamically unfavorable owing to the significantly lower reduction potential of the pair $2\ H^+/H_2$ (−0.42 V) compared to NAD^+/NADH (−0.32 V).

In 2009, the same year the model was published, it was discovered that in *T. maritima*, electrons are transferred from NADH to H^+ in a synergistic process involving a

ferredoxin with the much lower reduction potential of −0.453 V as a second electron donor [219]. The hydrogenase complex catalyzes the net reaction

$$\text{NADH} + 2\ \text{ferredoxin[red]} + 3\ \text{H}^+ \rightarrow \text{NAD}^+ + 2\ \text{ferredoxin[ox]} + 2\ \text{H}_2\ [219].$$

In the model, the two hydrogenase reactions are not directly coupled. As shown in Figure 19, their flux ratio is close to 1:1, however.

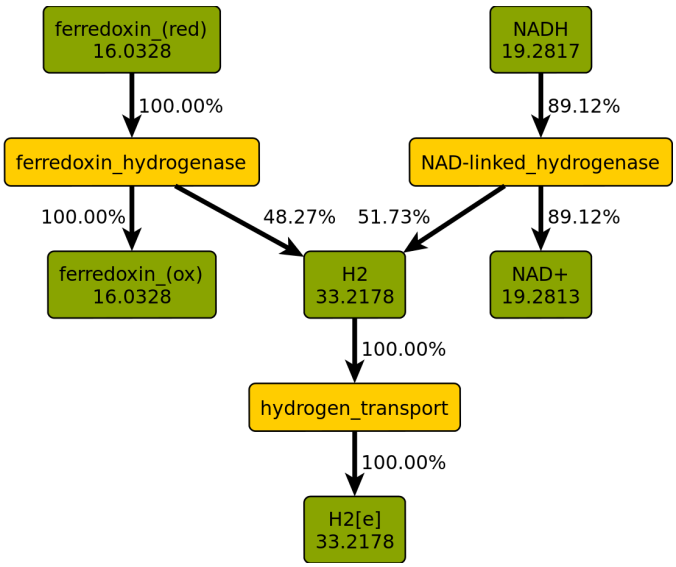

Figure 19: Flux split ratios of the two hydrogen-forming reactions in the FBA solution for T. maritima. Green: Metabolite nodes, yellow: enzyme nodes. Numbers in metabolite nodes are total flux in mmol gDW^{-1} h^{-1}. Edge labels are split ratios as fractions of the flux through the adjacent metabolite node.

Reduced ferredoxin is produced in the conversion of pyruvate to acetyl-CoA, which is catalyzed by pyruvate synthase (EC 1.2.7.1):

$$\text{coenzyme A} + \text{ferredoxin[ox]} + \text{pyruvate} \rightleftharpoons \text{acetyl-CoA} + \text{CO}_2 + \text{ferredoxin[red]} + \text{H}^+.$$

Table 19 shows the NADPH-producing reactions along with their flux split ratios. At 1.5 mmol gDW^{-1} h^{-1}, the total flux through NADPH is only 7.7% of the NADH flux (*E. coli* iJO1366: 41.4%). In the FBA solution, 48.5% of the NADPH is produced in the oxidative pentose phosphate pathway, while 26.9% is generated by isocitrate dehydrogenase in the formation of glutamate from citrate. 24.6% arises as a byproduct in purine metabolism.

The high relative contribution of purine metabolism to NADPH production is most likely due to the low overall flux through NADPH, which in turn is induced by the low demand for NADPH in the model: Glutamate dehydrogenase, which accounts for over 50% of the NADPH demand in *E. coli* is replaced by an NADH-dependent form in the *T. maritima* model. In addition, the amino acids threonine and methionine, whose bio-

syntheses account for a substantial fraction of the NADPH consumption in *E. coli*, are synthesized via non-NADPH-dependent pathways in the *T. maritima* model.

Table 19: NADPH balance (T. maritima). Total flux: 1.5 mmol gDW^{-1} h^{-1}. Ratios below 0.05% not shown.

Producing fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
isocitrate dehydrogenase	TCA cycle	26.90%
methylenetetrahydrofolate dehydrogenase	purines	24.60%
glucose 6-phosphate dehydrogenase	pentose phosphate pathway	24.25%
phosphogluconate dehydrogenase	pentose phosphate pathway	24.25%
Consuming fluxes		
Reaction/Enzyme	**Pathway/Product**	**Ratio**
ketol-acid reductoisomerase (2 reactions)	valine, leucine, isoleucine, coenzyme A	26.87%
thioredoxin reductase	synthesis of deoxyribonucleotides	18.66%
shikimate dehydrogenase	shikimate pathway	9.96%
fatty acid biosynthesis (4 reactions)	fatty acids	9.14%
aspartate semialdehyde dehydrogenase	lysine	8.57%
N-acetyl-γ-glutamyl-phosphate reductase	arginine, polyamines	7.83%
sulfide dehydrogenase	sulfur metabolism	5.88%
glutamate 5-semialdehyde dehydrogenase	proline	5.09%
pyrroline 5-carboxylate reductase	proline	5.09%
dihydrofolate reductase	folate metabolism	1.21%
UDP-N-acetylenolpyruvoylglucosamine reductase	peptidoglycans	0.67%
thymidylate synthase	dTMP	0.60%

Furthermore, the fraction of fatty acids per gram dry weight seems to be much lower in the *T. maritima* model than in *E. coli*. Like in *E. coli*, NADPH is used as a reducing agent in a wide variety of anabolic pathways: 60.6% is consumed in amino acid biosynthesis, 19.3% in the provision of deoxyribonucleotides, 9.1% in fatty acid biosynthesis, 5.9% in the reduction of sulfur, 2.4% in folate metabolism, and 1.0% in the formation of polyamines. Smaller fractions are consumed in the biosyntheses of isoprenoid precursors and coenzyme A.

5.2.3 Biosyntheses involving reversed degradation pathways

In the FBA solution, a conspicuously large amount of ATP is used to phosphorylate dGDP to dGTP. The reason for this was discovered by exploring the fate of dGTP via branch point analysis: While 3.7% enters the biomass as a building block of DNA, 96.3% is dephosphorylated in two steps to deoxyguanosine, which is then converted to

deoxyribose 1-phosphate and guanine via purine-nucleoside phosphorylase (EC 2.4.2.1). While the guanine is salvaged into GMP via guanine phosphoribosyl-transferase (EC 2.4.2.8), the deoxyribose 1-phosphate is converted to glyceraldehyde 3-phosphate and acetaldehyde via the action of phosphopentomutase (EC 5.4.2.7) and deoxyribose-phosphate aldolase (EC 4.1.2.4). The acetaldehyde is then used by threonine aldolase (EC 4.1.2.48) to convert glycine to threonine.

Threonine aldolase is known to be involved in the degradation of threonine [200]. From *in-vitro* studies, the corresponding reaction is known to be reversible [220]. However, no organism is known to rely exclusively on this reaction for the provision of threonine, and no biosynthetic pathway involving threonine aldolase has been described. The structure and substrate specificity of the threonine aldolase of *T. maritima* have been studied in detail. The corresponding publication [221] reports high pairwise sequence identities to threonine aldolases from organisms such as *E. coli*, which are known to produce threonine from homoserine via the usual biosynthetic pathway (see [200] for a detailed description).

In the model, threonine aldolase is the only possible way to produce threonine – if its flux is restricted to the cleavage direction, a non-zero biomass flux is no longer possible. In view of the current knowledge about threonine aldolase, both general and specific to *T. maritima*, this seems to be an error in the model. In the modeling process, it would have been much more parsimonious to assume the standard biosynthetic pathway even in the absence of specific evidence than to postulate a novel route of biosynthesis, especially if that route incorporates an enzyme that has highly similar homologs known to be active in a degradation pathway.

An exploratory series of branch point analyses starting from the balance sheets of ATP (Table 17) and ammonium (not shown) revealed several degradation reactions, which are used in biosynthetic routes in the model. Table 20 lists these reactions.

Table 20: Reactions known to occur in degradation pathways but involved in the biosynthesis of biomass precursors in the FBA solution for T. maritima. Reactions are written in the degradation direction. Reactions marked with a dagger (†) are the only possible way in the model to produce the respective compound.

Enzyme (named as in model)	EC number	Reaction
threonine aldolase †	4.1.2.48	L-threonine ⇌ glycine + acetaldehyde
L-serine deaminase †	4.3.1.17	L-serine ⇌ pyruvate + ammonium
L-aspartase	4.3.1.1	L-aspartate ⇌ fumarate + ammonium
O-succinylhomoserine lyase (elimination)	2.5.1.48	O-succinyl-L-homoserine + H_2O ⇌ 2-oxobutanoate + H^+ + ammonium + succinate
dCMP deaminase	3.5.4.12	dCMP + H^+ + H_2O ⇌ dUMP + ammonium
glucosamine-6-phosphate deaminase	3.5.99.6	D-glucosamine 6-phosphate + H_2O ⇌ D-fructose 6-phosphate + ammonium

The only way for the model to produce the proteinogenic amino acid serine is by using the degradation reaction catalyzed by L-serine deaminase (EC 4.3.1.17) in reverse. Serine biosynthesis usually starts from 3-phosphoglycerate and proceeds via the intermediate 3-phosphohydroxypyruvate (see [200] for a detailed description). An alternative non-phosphorylated route had been discussed for a few decades, but was ultimately dismissed as a 'fallacy' [222]. Consequently, serine biosynthesis via L-serine deaminase is likely an error in the model. L-Serine deaminase is the only way for the model to produce serine, as confirmed by an FBA performed with the flux through this reaction constrained to the deamination direction. The unusual biosynthetic pathways of serine and threonine are not discussed by the authors of the model.

In addition, the model contains 'shortcuts' for the production of aspartate, O-succinyl-L-homoserine, dCMP, and D-glucosamine 6-phosphate that involve degradation reactions. If these reactions (shown in Table 20) are blocked by appropriate flux constraints, the biomass flux decreases slightly (by 0.2% when constraining all four reactions), but biomass production is still possible.

In the model, aspartate is produced preferentially from fumarate and ammonium via aspartase, which is known to occur primarily in the degradation of aspartate. Aspartate is usually produced from oxaloacetate via aspartate transaminase, which uses glutamate as the amino group donor [200]. This reaction is more costly, as the regeneration of glutamate from 2-oxoglutarate consumes NADPH or NADH, respectively (see the first two subsections in Section 5.1.2).

Similarly, O-succinyl-L-homoserine is produced preferentially via an elimination reaction catalyzed by O-succinyl-L-homoserine succinate-lyase (EC 2.5.1.48, synonym: cystathionine gamma-synthase). In the reaction, which is shown in Table 20, O-succinyl-L-homoserine is cleaved into 2-oxobutanoate, succinate, and ammonia. However, in the model, O-succinyl-L-homoserine can also be formed from the three smaller molecules in the reverse reaction, which seems unlikely from both enthalpic and entropic considerations. The reaction, which occurs during the biosynthesis of methionine, is a cheaper alternative to the standard reactions, which produce O-succinyl-L-homoserine or alternatively O-acetyl-L-homoserine from L-homoserine and succinyl-CoA or acetyl-CoA, respectively.

In the model, dCMP is formed from dUMP and ammonium via dCMP deaminase, and D-glucosamine 6-phosphate is produced by glucosamine-6-phosphate deaminase from D-fructose 6-phosphate and ammonium. These two reactions also bypass energy-dependent biosyntheses. If dCMP deaminase is constrained to the degradation direction, dCDP is formed from CDP at the expense of thioredoxin oxidation in a reaction catalyzed by ribonucleoside-diphosphate reductase. Similarly, if the reaction of glucosamine-6-phosphate deaminase is constrained to the degradation direction, D-glucosamine 6-phosphate is generated from D-fructose 6-phosphate via a transaminase using glutamine as the amino-group donor.

The impact of these modeling errors is twofold: The reversibility of the degradation reactions for serine and threonine obfuscates the fact that the biosyntheses of these two

proteinogenic amino acids have not been modeled. The reversibility of the other four reactions listed in Table 20 introduces unlikely alternative reactions to enzymes known (from other organisms) to be involved in the biosyntheses of the respective compounds. In addition, they lead to quantitative effects in energy metabolism, lowering the demand for ATP, NADH, and NADPH per unit of biomass. These effects are low for the last three reactions as methionine, dCDP, and D-glucosamine 6-phosphate do not carry high fluxes at optimal biomass production. However, the flux through aspartate in the FBA solution is relatively high at 1.0 mmol gDW^{-1} h^{-1} (at a biomass flux of 0.36 h^{-1}). Together with the fact that the transamination of aspartate to asparagine and the energy demand of non-growth associated processes have been omitted from the model, this likely leads to a considerable underestimation of the cellular energy consumption by the model.

Because of the gaps identified in amino acid biosynthesis and the overly far-reaching metabolic capabilities of the model, a more detailed analysis was not deemed useful. In order for the model to make accurate predictions of the metabolic phenotype of *T. maritima* under different conditions, these issues have to be addressed through careful manual curation and refinement of the model.

5.3 Performance of wMOMA vs. MOMA on a small genome-scale model

A preliminary analysis was performed to assess the predictive quality of wMOMA in comparison to MOMA. To keep computation times low, the methods were applied to all single-reaction knockouts in a small genome-scale metabolic model. The model iMO1056 of *Pseudomonas aeruginosa* PAO1 [79] was chosen for this analysis. At 1,005 reactions and 874 metabolite nodes it is considerably smaller than any of the *E. coli* models. All simulations were performed with the minimal medium constraints provided in the same file as the model (aerobic growth on glucose, glucose uptake rate: 10 mmol gDW^{-1} h^{-1}, see Section 3.1.3). For this scenario, FBA predicted a specific growth rate of 1.05 h^{-1}. 385 metabolites are dead ends in this scenario, leading to 369 blocked reactions. Using the automated knockout analysis implemented in *metano*, the growth rate of each single-reaction knockout mutant was predicted by FBA, MOMA, and wMOMA in turn. As noted in Section 4.5, FBA gives an upper limit for the biomass flux that is possible in the mutant network.

An overview of the results of this preliminary analysis is shown in Figure 20. For the 636 reactions that were not blocked due to dead ends (and thus predicted to be nonessential by FBA, MOMA, and wMOMA alike), the predictions of FBA differed strongest from those of the other methods. All three methods predicted the knockouts of about 230 reactions (FBA: 229, MOMA: 236, wMOMA: 234) to be definitely lethal (biomass flux < 1% of the wild-type optimum). In addition, the knockouts of more than 340 reactions (FBA: 367, MOMA: 342, wMOMA: 348) were predicted by all three methods to have no effect on the growth rate (biomass flux > 99% of the wild-type optimum).

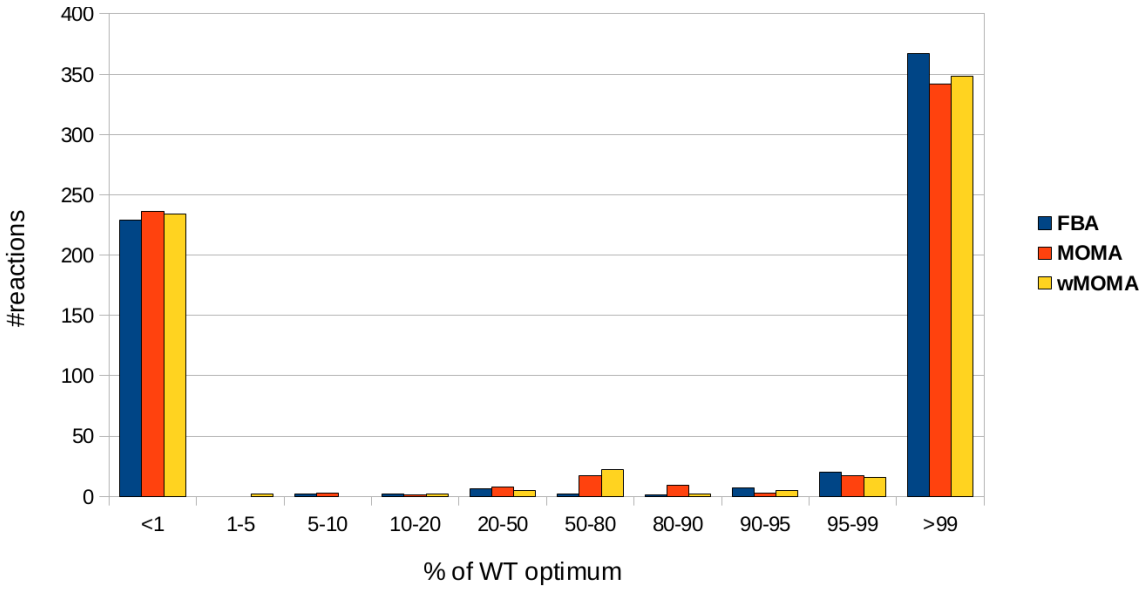

Figure 20: Histogram of the predicted biomass flux for each single-reaction knockout relative to FBA optimum for the wild-type. 369 reactions that are blocked due to dead ends were excluded.

The predictions of MOMA and wMOMA with respect to the biomass flux differed for very few cases only, as can be seen in Figure 21. The difference was less than 10^{-6} % of the wild-type optimum for the knockouts of 533 of the 636 non-blocked reactions. A difference of more than 10% of the wild-type optimum was found for only 12 single-reaction knockouts. Of these, a difference of more than 20% (but less than 50%) was found for only 3 cases: proton-translocating NADH dehydrogenase and the proton-translocating cytochrome complexes cytochrome c oxidase and ubiquinone-cytochrome c reductase. Unfortunately, experimental information about these knockouts, including whether the mutants are viable at all, was not available. As all three of these proteins are involved in the respiratory chain and *P. aeruginosa* is an obligate respirer [144], the effects of the mutations are likely pronounced.

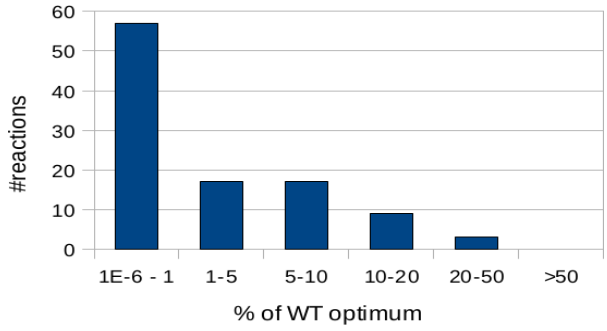

Figure 21: Histogram of the difference in predicted biomass flux between MOMA and wMOMA expressed as a percentage of the wild-type FBA optimum. Scenarios with a difference below 10^{-6} % not shown.

As MOMA and wMOMA make very similar predictions for most scenarios, the quality of their predictions cannot be easily evaluated. In particular, such an evaluation would require a model that has been very thoroughly validated and refined. Moreover, high-quality *in-vivo* flux data are needed. In addition to differences in the predicted biomass flux, further criteria could be used to identify knockouts for which MOMA and wMOMA make qualitatively different predictions. Examples of such criteria are differences in the predicted CO_2 secretion rate, the summed absolute flux through all reactions in the network, and the ratio of CO_2 export and biomass flux. Furthermore, flux distributions predicted by MOMA and wMOMA for the same scenario could be classified as qualitatively different if a reaction flux has opposite signs in the two flux distributions or if a reaction is active in one flux distribution but inactive in the other. The question which method makes the better predictions for specific scenarios will have to be explored in a follow-up study. Nevertheless, wMOMA is more theoretically sound than MOMA, as fluxes that cannot be accurately predicted via FBA (as evidenced by their very high variability in FVA) do not need to be faithfully reproduced by the mutant network in wMOMA, in contrast to MOMA.

6 *metano* as a tool for model reconstruction

While the preceding chapters focused on the uses of *metano* in the analysis of metabolic models and predicted flux distributions, the present chapter will outline the roles that *metano* can play in the process of model construction and refinement. Figure 22 gives an overview of the iterative process of the reconstruction of genome-scale metabolic networks. This cycle can be seen as a specialization of the more general cycle of data- and hypothesis-driven modeling in systems biology shown in Figure 1 in Section 2.1 (page 6).

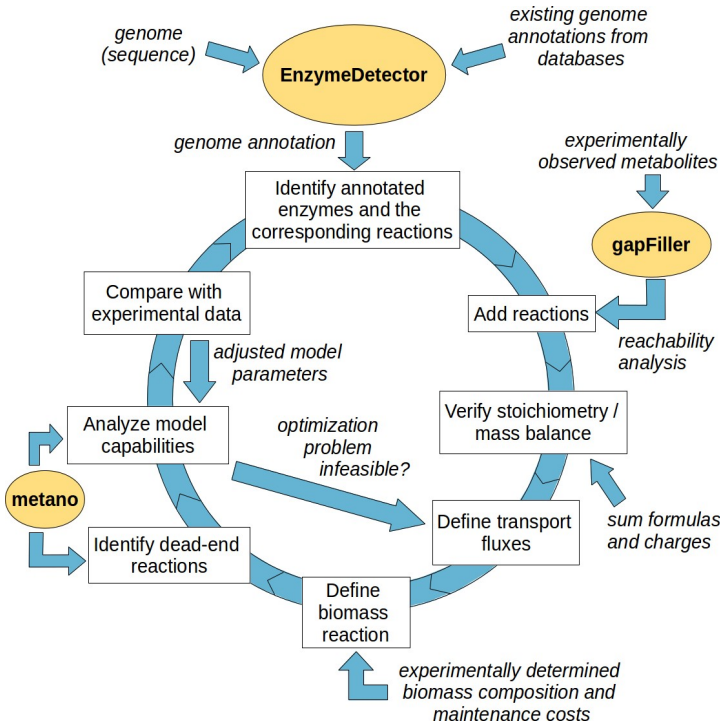

Figure 22: The roles of metano and other software tools developed in our group in the iterative cycle of genome-based metabolic modeling.

6.1 Background: From the genome to a network model

As the tools implemented in *metano* operate only on stoichiometric metabolic models, they are not applicable in the first stages of network reconstruction. To provide a rough outline of the modeling process, these early stages will briefly be described in this section. These stages correspond to the right-hand side of the cycle shown in Figure 22.

As noted in Section 2.5.1, the process of metabolic modeling always starts from a genome annotation. This annotation is usually compiled by combining various

annotation sources. The program EnzymeDetector, which was developed in our group, combines data from different biological databases with *in-silico* predictions of gene functions [105]. The computational methods used for enzyme function prediction include a BLAST-based homology search [105] and the hierarchical pattern-based function assignment BrEPS [109].

In the next step, a list of catalyzed reactions is created from the list of annotated enzyme functions. Different electronic databases can be used for the assignment of biochemical reactions to enzyme functions. The database BKM-react [223], which was developed in our group, is an integrated database of biochemical reactions compiled from the databases BRENDA [106], KEGG [107], and MetaCyc [224]. Based on the initial list of metabolic reactions, both enzyme-catalyzed and spontaneous, a raw model can be constructed. In the construction of stoichiometric models, it is important that all reactions satisfy the condition of mass balance (see Section 2.3). As the stoichiometry of each reaction included in BKM-react has been verified [223], this database is particularly well suited for metabolic modeling.

The raw model can be supplemented with reactions that are required for the reachability of known biomass components. Different strategies exist for this process, which has been termed ‘gap filling’, including graph-based approaches and mathematical optimization strategies. The program gapFiller, which was developed in our group, exemplifies the graph-based strategy [225], while the MetaFlux tool employs mixed-integer linear programming [226].

In order to obtain a complete ‘simulation-ready’ metabolic model, compartments have to be defined, and transport and exchange reactions have to be added to the model (see Section 2.3.3). In addition, a biomass reaction has to be formulated (see Section 2.3.5). To date, these steps cannot be performed in an automated fashion.

6.2 Getting the model to ‘run’

Once an initial network model with biomass, transport, and exchange reactions has been constructed, *metano* can be used to analyze the model’s capabilities and to fit model parameters. In the first stage, the model is refined with the goal to obtain a steady-state flux distribution, i.e. to make the **FBA** optimization problem feasible. The problem is infeasible, i.e. the solution space is empty, if no flux vector exists that simultaneously satisfies all equality and inequality constraints. For instance, the substrate or oxygen uptake rate might be too low to satisfy the non-growth-associated maintenance requirement (NGAM). Other common causes of infeasibility are missing transport or exchange reactions and gaps in pathways leading to biomass precursors. At this stage, **dead-end analysis** can be a valuable tool for identifying gaps in the network. If metabolites that are known intermediates in a biosynthetic pathway are dead ends in the model, this might indicate a gap in the pathway. Dead-end analysis can also identify problems resulting from inconsistent use of metabolite identifiers (e.g. if molecular oxygen is labeled ‘oxygen’ in some reactions and ‘O2’ in others).

A further common occurrence at this stage of modeling is that while the solution space of **FBA** is not empty, a non-zero biomass flux is not possible. This is usually the result of unintended stoichiometric coupling between reactions. An example is shown in Figure 23.

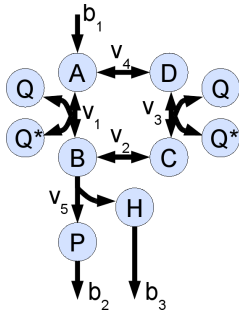

Figure 23: An example of stoichiometric coupling: In this network, the reactions labeled 'v_1' and 'v_3' are stoichiometrically coupled via the cofactor metabolites Q and Q.*

In this network, the reactions labeled with the flux variables v_1 and v_3 are stoichiometrically coupled so that, depending on notation, either $v_1 = v_3$ or $v_1 = -v_3$, and $b_1 = v_5 = 0$. The reason for this is a missing pathway synthesizing or degrading either Q or Q*. As neither Q nor Q* can be produced or consumed outside the reactions v_1 and v_3, the two reaction fluxes must cancel each other out in steady state. Consequently, it is impossible to add matter to the cycle A–B–C–D via flux b_1 or remove matter via flux v_5. In practice, this phenomenon can occur, e.g., when cofactor biosynthetic pathways still contain gaps.

If it is possible for the objective function in **FBA** to assume a non-zero value, the solution space might not be bounded (except possibly for an artificial upper bound on the objective function). The cause for this is either a lack of constraints on substrate uptake rates or the presence of a reaction violating mass balance, which allows the production of matter out of nothing. The first case can easily be discovered by examining the boundary fluxes in the FBA solution. In the latter case, the problematic reaction can be identified by tracing back from any metabolite with an implausibly high flux using **split-ratio analysis** or its graphical extension branch point analysis (BPA).

Once a finite, non-zero biomass flux is obtained in FBA, the fitting of model parameters to thermodynamic constraints and experimental data can begin. If the physiological ranges of intracellular metabolite concentrations are known, they can be used for the **free-energy-based assignment of reaction directions**. In the **FBA** solution, boundary fluxes should be closely examined for plausibility. If an energy-rich metabolite is secreted, this might indicate an error in the model. Possibly, a reaction metabolizing the exported compound is missing in this case. Unexpected behavior can be traced back to its immediate cause using **split-ratio analysis** and/or BPA. At this stage, **FVA** can also be useful for the identification of overly far-reaching capabilities by examining for each flux variable whether it can be zero and whether it must be zero in the studied scenario. For instance, in a model of an obligate aerobe, growth should be dependent on a non-zero oxygen uptake rate.

6.3 Model refinement

Metabolic modeling is a cyclic process (see Figure 22), where new reactions are added as new metabolic functions are found – either in an updated genome annotation or by experimental observation. By monitoring assertions via *metano*'s tool for **automated plausibility checking** at every step of the modeling process, it is possible to determine if a change to the model leads to a qualitative change in the predicted metabolic behavior. Seemingly small changes in the model can alter the model's predictions dramatically. For instance, if a transporter is introduced that is not constrained, unlimited substrate uptake and thus unlimited growth may be possible. To give another example, a single added reaction may lead to a biologically infeasible 'shortcut' in existing pathways.

Finally, systematic metabolite-centric analyses like those described in Sections 5.1 and 5.2 can be used to fit model predictions to experimental results.

7 Summary and conclusion

In this work, a collection of software tools for the reconstruction and analysis of metabolic models has been developed. The software package *metano* implements the established computational methods of flux balance analysis (FBA), flux variability analysis (FVA), minimization of metabolic adjustment (MOMA), dead-end analysis, and the free-energy-based prediction of reaction directionality. Within this framework, several new methods and extensions of existing algorithms have been implemented, to wit weighted MOMA (wMOMA), split-ratio analysis, metabolite flux minimization (MFM), and the automated plausibility checking of metabolic models. With its broad selection of computational methods, *metano* is a useful tool to assist in the process of metabolic reconstruction and to perform in-depth analyses of metabolic models and their predicted behavior in different growth scenarios.

The distribution of *metano* as open source allows anyone to use the software (under the terms of the GNU General Public License) and to adapt it to their specific requirements. In this way, *metano* serves as a reference implementation of the algorithms developed in this work. The object-oriented design of the software ensures easy maintainability and extensibility. *metano* has been implemented in such a way as to be easy to use for users with different technical backgrounds: On the one hand, it is written in the widely-used Python programming language and provided in the form of Python classes and functions. This permits easy adaptation of the software to different requirements by users with basic programming skills. On the other hand, *metano* is provided in the form of stand-alone Python programs, and it supports the popular SBML format for the exchange of metabolic models. These features allow metabolic modelers without programming experience to quickly apply the methods implemented in *metano* to their models.

In the design of *metano*, great attention was paid to efficiency. In the implementations of FBA, FVA, MOMA/wMOMA, and MFM, iterative dead-end analysis is performed as a preprocessing step in order to reduce the dimension of the solution space of the respective optimization problems. The implementations of FVA and MFM, which optimize different objective functions in the same solution space, employ the ‘fastFVA’ strategy, where each optimization is launched from the solution of the previous one. MOMA/wMOMA have been implemented with the option to perform an FVA-based reduction of the steady-state solution space as an alternative preprocessing step, which can speed up the analysis even more than a dead-end analysis. Due to its good run-time behavior, *metano* has been successfully applied to large-scale analyses involving thousands or even hundreds of thousands of simulations [187,197].

The implementation of FBA in *metano* is one of the first to allow unbounded flux variables in the optimization. As has been argued in Section 4.3 and demonstrated on the example of the *E. coli* model iAF1260, unproductive fluxes through stoichiometrically balanced cycles do not occur in this implementation. *metano* FBA hence does not

require special treatment of thermodynamically infeasible loops. Therefore, it is more computationally efficient than existing implementations, which often use mixed-integer linear programming to secondarily preclude cyclic flux through such loops.

Weighted MOMA (wMOMA) is a variant of MOMA, which is introduced in this thesis and has been implemented in *metano*. In contrast to MOMA, it is much less sensitive to the choice of reference vector from the optimal FBA solution space. While the preliminary analysis described in Section 5.3 failed to clearly identify an experimentally verifiable scenario where the two methods make substantially different predictions, wMOMA is more theoretically sound than MOMA. FBA returns arbitrary values for variables that are not sharply constrained by the optimization problem. Using such arbitrary values as reference is obviously always wrong.

metano uses the same data structures and functions for both MOMA and wMOMA. As most of the code is shared between the two methods, improvements in one method become available in the other instantaneously. This shared code is easier to maintain than duplicated code. The use of shared code is made possible by the fact that the two methods differ only in the weighting scheme (uniform in MOMA and linear in wMOMA).

In this work, it has further been shown how the automated plausibility checking of metabolic models is made possible by the formalization of expectations with respect to model behavior. The formalization proposed here takes the form of assertions, i.e. Boolean expressions which evaluate to `True` for the expected case. A procedure for sequentially testing a series of assertions has been implemented in *metano*. The resulting program can be used in the process of model construction and refinement to ensure that any unexpected model behavior enabled by a modification or extension of the model is discovered as soon as it arises.

The methods of split-ratio analysis and metabolite flux minimization, which were developed in this work, establish a metabolite-centric view on flux distributions. This metabolite-centric view allows a more meaningful biological interpretation of flux distributions than is possible from the reaction fluxes alone. It thus complements the classical reaction-centric framework.

The detailed analyses presented in Section 5.1.2 serve to illustrate the power of split-ratio analysis and its graphical extension branch point analysis (BPA): While split-ratio analysis merely post-processes an FBA solution, it yields all split ratios and metabolite fluxes in the system at a glance. Both the tabular output of split-ratio analysis and, in particular, the interactive graphical output of BPA allow analyzing reaction fluxes in a natural context. The context for interpreting the flux through any particular reaction is provided by considering not the absolute flux value in mmol gDW^{-1} h^{-1}, but rather its share of the total flux through each of the metabolites consumed or produced by the reaction.

Split-ratio analysis and BPA allow the scenario-specific assignment of metabolic roles: For instance, the energy balance of *Escherichia coli* cannot be fully understood without

considering phosphoenolpyruvate (PEP). In contrast, in *Thermotoga maritima*, PEP mostly acts as a catabolic intermediate, which is used to generate pyruvate and ATP. Using split-ratio analysis, metabolites can be classified, e.g., as mere biomass components, energy or redox carriers, group donors, and mere byproducts. To give another example, in *T. maritima*, the reducing equivalent NADH acts mostly as a catabolic byproduct, as its chemical potential is used for the largest part to reduce intracellular protons to hydrogen, which does not contribute to biomass formation.

The metabolite-centric analyses of the two very different scenarios of aerobically growing *E. coli* and strictly anaerobic *T. maritima* illustrate that the energy metabolism under a particular set of environmental conditions can be characterized in terms of the balance sheets of a very small number of currency metabolites. These balance sheets can be used to assess the biological plausibility of a model's predictions even when organism-specific experimental data are scarce. This is due to the fact that energy metabolism tends to proceed along variations of only a few well-understood pathways, depending almost exclusively on the organism's ecology (phototrophic vs. chemotrophic, anaerobic vs. aerobic, etc.).

Moreover, as only a fraction of the reactions in a model is active in any particular scenario and most of the active metabolite nodes are purely transitory, a full metabolite-centric analysis will likely never need to consider more than a few hundred metabolite nodes.

By studying the split ratios of branch points at the intersection of catabolism and anabolism, such as 2-oxoglutarate and oxaloacetate, it is possible to explore the balance between energy generation and biomass production for each pathway. Shifting this balance is crucial in metabolic engineering: If the goal is the overproduction and secretion of a metabolite of interest, it is not sufficient to overexpress enzymes on the pathway leading to product formation. Rather, an optimal balance has to be found between product formation, energy generation, and growth [227,228]. Thus, split-ratio analysis can be used to guide the process of metabolic engineering.

The analyses of metabolite pools and metabolic cycles presented in Section 5.1.3 are not easily possible with other methods than branch point analysis: The presence of cyclic fluxes is neither evident in the reaction fluxes nor in the balance sheets of single metabolites, but can only be seen in a larger metabolic context. Similarly, metabolite pools, which are depleted and replenished via various reactions from different pathways can only be understood in a larger metabolic context. This context is defined by all activated and depleted forms of a pool metabolite, which have to be examined simultaneously along with all reactions in which they are involved. This is powerfully illustrated by the analysis of the tetrahydrofolate (THFA) pool, which is defined by the activated forms 5,10-methylene-THFA, 5-methyl-THFA, 5,10-methenyl-THFA, and 10-formyl-THFA and the depleted forms THFA and dihydrofolate.

Metabolite flux minimization is a second metabolite-centric method developed in this work and implemented in *metano*. As demonstrated in Section 5.1.4, it is a useful tool for efficiently assessing the essentiality of metabolite nodes for optimal biomass

production. In addition, a ranking of metabolite nodes by their required flux under the goal of optimal biomass production can give further insights into the global metabolic behavior. Studies of metabolite essentiality are not possible in a purely reaction-centric framework. In the example of the *E. coli* model iJO1366, it is not apparent from the flux variabilities of the ten reactions involving thioredoxin or the twenty-three reactions involving menaquinone-8 that flux through neither of these metabolites is required for optimal biomass production in the studied scenario.

The detailed metabolite-centric analyses of the published genome-scale metabolic models iJO1366 and iTZ478 presented in Sections 5.1 and 5.2 revealed several prospects for model refinement. From the balance sheets of energy metabolites, it was discovered that in the *E. coli* model iJO1366, the reaction pair fructose-6-phosphate (F6P) aldolase + dihydroxyacetone kinase can partially replace the usual reaction pair of phosphofructokinase + fructose-bisphosphate aldolase in glycolysis. As this involves activity of F6P aldolase in the strongly endergonic cleavage direction and activity of this enzyme in either direction has never been observed *in vivo*, this alternative route in glycolysis should be disallowed in the model. This would remove ambiguity in the model's predictions and thus improve accuracy for most scenarios.

In the systematic survey of the split ratios of all energy metabolites, a discrepancy was found between the predicted and experimentally determined NADPH balance for *E. coli* growing aerobically on glucose. The discrepancy could be resolved completely by introducing a forced flux through proton-translocating transhydrogenase.

Two further possibilities for refinement of the model iJO1366 were identified in an analysis of metabolite essentiality using metabolite flux minimization. Firstly, it was discovered that in the model, excess intracellular protons can be converted to water using molecular oxygen and electrons donated by Fe^{2+} ions, which are oxidized to Fe^{3+} in the process. As this entails iron uptake and secretion at an extreme rate and the solubility of the produced Fe^{3+} ions is low, this reaction should also be blocked in the model to eliminate ambiguity and improve accuracy. Secondly, fructose 1,6-bisphosphate is predicted to be nonessential in the model for optimal biomass production under aerobic growth on glucose. The reason was found in the sedoheptulose bisphosphate bypass, which can completely replace the standard reactions of phosphofructokinase and fructose-bisphosphate aldolase. However, this alternative metabolic route has only been described for transaldolase-deficient mutants, and it uses the same enzymes as the standard glycolytic path. Therefore, it should be disallowed when specifically studying the wild type.

As the authors of the *T. maritima* model iTZ478 themselves note, the model is not accurate enough to allow quantitative predictions. Obvious limitations are that non-growth-associated processes are not represented in the model, that the formation of asparagine has not been modeled, and that the value of the substrate uptake rate seems to have been selected arbitrarily. By analyzing the split ratios of high-energy metabolites such as ATP and small low-energy metabolites such as ammonium as predicted for the studied scenario, further unexpected metabolic capabilities were discovered: Several reactions known only from degradation pathways were found to be

involved in the biosynthesis of biomass precursors, where they operate in the reverse direction. In particular, the proteinogenic amino acids serine and threonine can only be produced via reversed degradation pathways. As no *in-vivo* evidence exists for any of these reversed pathways, they must be considered errors in the model. Four other degradation reactions operating in the reverse direction were found to act as 'shortcuts', which bypass the standard biosynthetic pathways.

These results illustrate that the metabolite-centric analysis of flux distributions made possible by algorithms implemented in *metano* is a new and powerful tool for the reconstruction and refinement of genome-scale metabolic models. Using the metabolite-centric framework developed in this work, possibilities for model refinement could be identified even in the model iJO1366, which is the result of more than a decade of genome annotation and metabolic modeling.

Acknowledgments

This work has benefited greatly from the support of many people to whom I am very grateful. First of all, I would like to express my gratitude to Prof. Schomburg for supervising this thesis, for giving me the opportunity to work in the interesting field of systems biology, and for helpful discussions during all stages of the work. I would also like to thank Prof. Jahn for agreeing to review this thesis.

I am very grateful to Dr. Thomas Ulas, Thomas Riemer, and Dr. Kerstin Schmidt-Hohagen for taking time out of their busy schedules to proofread parts of this thesis. Furthermore, I would like to thank Dr. Antje Chang, Dr. Maurice Scheer, and Dr. Ida Schomburg for sharing their experience with the intricacies of scientific publishing.

I would like to thank Dr. René Rex for helpful suggestions and hands-on assistance in the implementation in *metano*. I am thankful to Adam Podstawka, Sandra Placzek, and Christian Cholewa for providing technical assistance and to Adam Podstawka for setting up the *metano* website. I am also grateful to Chitrarajkumari Balasubramanian for taking up the task of systematically evaluating wMOMA in comparison with regular MOMA.

Good software is made better by user feedback. I am greatly indebted to the modelers in our group, particularly Dr. Thomas Ulas, Dr. René Rex, Ingrid Hartwig, Astrid Fieselmann, and Helge Stark, for testing *metano* and providing helpful feedback. I would further like to thank the many students who have attended practical courses under my supervision for providing feedback on the user experience of *metano* for learners and for allowing me to develop my teaching skills.

I am grateful to everyone in the Department of Bioinformatics and Biochemistry, now and when I started my work, for the many interesting conversations and for always providing a pleasant and stimulating work atmosphere. Finally, I would like to thank my family and friends for the support and encouragement they provided during the time of my work on this thesis.

Symbols and abbreviations

gDW	Gram dry weight of biomass
lb	Vector of lower bounds on the reaction fluxes
ub	Vector of upper bounds on the reaction fluxes
S	Stoichiometric matrix
v	Vector of reaction fluxes
v_{BM}	Biomass flux (flux through the virtual biomass reaction)
Z**(v)**	A function in the fluxes $\boldsymbol{v}$ used as objective function in FBA
$Z_{opt, FBA}$	The optimal value of the objective function determined by FBA

ABC transporter	ATP-binding cassette transporter
BPA	Branch point analysis
CoA	Coenzyme A
DHA	Dihydroxyacetone
F6P	Fructose 6-phosphate
FBA	Flux balance analysis
FVA	Flux variability analysis
G6P	Glucose 6-phosphate
GAM	Growth-associated maintenance requirement
MFM	Metabolite flux minimization
MOMA	Minimization of metabolic adjustment
NGAM	Non-growth-associated maintenance requirement
PEP	Phosphoenolpyruvate
PPP	Pentose phosphate pathway
PTS	Phosphotransferase system
ROOM	Regulatory on/off minimization
SBML	Systems biology markup language
TCA cycle	Tricarboxylic acid cycle
THFA	Tetrahydrofolate
wMOMA	Weighted minimization of metabolic adjustment

List of Figures

List of Tables

Appendix A: Comparison of FBA solutions from different software tools

FBA was performed on the *E. coli* model iAF1260 for scenario 1 (glucose uptake rate: 8 mmol gDW^{-1} h^{-1}, oxygen uptake rate: ≤ 18.5 mmol gDW^{-1} h^{-1}; see Section 3.1.1.1) using *metano* or SurreyFBA [185] (the latter via its JyMet graphical user interface). Both FBA solutions were then compared to the flux distribution published by the authors of iAF1260 (downloaded from the supplementary material of the publication [24]), which was generated by the FBA implementation in the COBRA toolbox [14].

The three FBA solutions agree for 2,301 of the 2,382 reactions in the network, which includes 234 reactions blocked due to dead ends. Only slightly more than 400 reactions are active in any FBA solution (at a cutoff of 10^{-12} mmol gDW^{-1} h^{-1}; *metano*: 403, COBRA toolbox: 403, SurreyFBA: 416). All three implementations predict the same biomass flux of 0.7367 h^{-1}, which means that all three solutions are indeed optimal (agreement within 10^{-6} h^{-1}; COBRA toolbox and SurreyFBA solutions are rounded to six digits). All three software tools predict an oxygen exchange flux well above the lower bound of −18.5 mmol gDW^{-1} h^{-1} defined in the scenario, i.e. growth is only limited by carbon availability and not by oxygen supply. There is no export of acetate in any of the three FBA solutions.

The 81 reactions for which differences were present between at least two flux distributions are shown in Table 21. In contrast to the other tools, SurreyFBA predicted extremely high fluxes for 19 reactions in infeasible loops (first 19 rows in Table 21, marked in red). Cyclic fluxes through infeasible loops do not occur in the solutions of *metano* and the COBRA toolbox. While for 81 reactions there is no agreement between all three tools, there are only 10 cases where no two solutions agree. The predictions of *metano* agree in 25 cases with the COBRA toolbox (marked in purple in Table 21) and in 22 cases with SurreyFBA (marked in green), while SurreyFBA and the COBRA toolbox agree for 23 reactions (marked in yellow). The differences are explained by alternative pathway use, towards which the FBA objective, maximization of biomass flux, is blind. To give an example, the following two sets of reactions are stoichiometrically equivalent:

I (active in COBRA toolbox solution):

R_phosphoribosylpyrophosphate_synthetase : ATP + ribose 5-phosphate → AMP + H^+ + phosphoribosyl pyrophosphate

R_adenylate_kinase : AMP + ATP → 2 ADP

II (active in *metano* solution):

R_ribose_1_5_bisphosphokinase : ATP + ribose 1,5-bisphosphate → ADP + phosphoribosyl pyrophosphate

R_ribose_1_phosphokinase : ATP + ribose 1-phosphate → ADP + H^+ + ribose 1,5-bisphosphate

R_phosphopentomutase : ribose 1-phosphate ← ribose 5-phosphate

Table 21: Differences in the flux distributions predicted by three different implementations of FBA (metano, COBRA toolbox, SurreyFBA) for the model iAF1260, scenario 1 (see Section 3.1.1.1). Flux values are in mmol gDW^{-1} h^{-1}. All three FBA solutions agree for any reaction not shown.

Reaction	*metano*	COBRA toolbox	SurreyFBA
R_adentylate_kinase__GTP_	0.9816	0	-999999
R_nucleoside_diphosphate_kinase__ATPGDP_	1.5420	0.5604	-999998
R_adenosine_transport_in_via_proton_symport__reversible__periplasm_	0	0	-999999
R_inosine_transport_in_via_proton_symport__periplasm_	0	0	999999
R_thymidine_transport_in_via_proton_symport__reversible__periplasm_	0	0	-999999
R_isochorismate_synthase	0	0	-999999
R_glucose_transport_via_diffusion__extracellular_to_periplasm_	8.0	8.0	-999991
R_Isochorismate_Synthase	0	0	999999
R_D_glucoseMaltotriose_transport_via_diffusion__extracellular_to_periplasm__irreversible	0	0	999999
R_inosine_transport_in_via_proton_symport__reversible__periplasm_	0	0	-999999
R_adenosine_transport_in_via_proton_symport__periplasm_	0	0	999999
R_thymidine_transport_in_via_proton_symport__periplasm_	0	0	999999
R_Valine_pyruvate_aminotransferase	0	0	999999
R_L_alanine_transaminase	-0.4294	-0.4294	-999999
R_phosphoribosylpyrophosphate_synthetase	0	0.6866	-999998
R_phosphopentomutase	-0.6866	0	-999999
R_ribose_1_phosphokinase	0.6863	0	999999
R_ribose_1_5_bisphosphokinase	0.6863	0	999999
R_valine_transaminase	-0.3118	-0.3118	999998
R_adenylate_kinase	0	1.6682	1.6682
R_fructose_6_phosphate_aldolase	1.0269	1.0269	0
R_pyruvate_kinase	0	0	1.0269
R_Dihydroxyacetone_phosphotransferase	1.0269	1.0269	0
R_fructose_bisphosphate_aldolase	5.1642	5.1642	6.1912
R_phosphofructokinase	5.1642	5.1642	6.1912
R_beta_ketoacyl_ACP_synthase__2_	0.2621	0.2621	0
R_Acetyl_CoA_ACP_transacylase	0	0	0.2621
R_beta_ketoacyl_ACP_synthase	0	0	0.2621
R_thioredoxin_reductase__NADPH_	0.2598	0	0.2598
R_fumarate_reductase_001	0	0.2437	0.2459

Reaction	*metano*	COBRA toolbox	SurreyFBA
R_succinate_dehydrogenase__irreversible_	3.7291	3.9750	3.9750
R_dihydroorotic_acid__menaquinone_8_	0	0.2437	0.2437
R_dihydoorotic_acid_dehydrogenase__quinone8_	0.2437	0	0
R_glutaredoxin_reductase	0	0.2206	0
R_glutathione_oxidoreductase	0	0.2206	0
R_phosphoadenylyl_sulfate_reductase__glutaredoxin_	0	0.1815	0
R_phosphoadenylyl_sulfate_reductase__thioredoxin_	0.1815	0	0.1815
R_flavodoxin_reductase__NADPH_	0	0.0392	0
R_ribonucleoside_triphosphate_reductase__CTP___flavodoxin_	0	0.0199	0
R_ribonucleoside_diphosphate_reductase__GDP___glutaredoxin_	0	0.0199	0
R_nucleoside_diphosphate_kinase__ATPCDP_	0.1134	0.1333	0.1134
R_nucleoside_diphosphate_kinase__ATPdCDP_	0.0199	0	0.0199
R_ribonucleoside_diphosphate_reductase__CDP_	0.0199	0	0.0199
R_ribonucleoside_diphosphate_reductase__GDP_	0.0199	0	0.0199
R_ribonucleoside_diphosphate_reductase__UDP_	0.0193	0	0.0193
R_ribonucleoside_diphosphate_reductase__ADP_	0.0193	0	0.0193
R_nucleoside_diphosphate_kinase__ATPdADP_	0.0193	0	0.0193
R_ribonucleoside_triphosphate_reductase__ATP___flavodoxin_	0	0.0193	0
R_ribonucleoside_diphosphate_reductase__UDP___glutaredoxin_	0	0.0193	0
R_iron__II__transport_via_diffusion__extracellular_to_periplasm_	0.0108	0.0056	0.0056
R_Fe2_exchange	-0.0108	-0.0056	-0.0056
R_Fe3_exchange	0	-0.0052	-0.0052
R_iron__III__transport_via_diffusion__extracellular_to_periplasm_	0	0.0052	0.0052
R_H_exchange	6.7609	6.7661	6.7661
R_proton_transport_via_diffusion__extracellular_to_periplasm_	-6.7609	-6.7661	-6.7661
R_glycolate_transport_via_sodium_symport__periplasm_	0	0.0035	0
R_glycolate_transport_via_proton_symport__reversible__periplasm_	0	-0.0035	0
R_acetate_reversible_transport_via_proton_symport__periplasm_	-0.0035	0	0
R_NaAcetate_symport__periplasm_	0.0035	0	0
R_L_proline_reversible_transport_via_proton_symport__periplasm_	0	0	-0.0035
R_NaProline_L_symporter__periplasm_	0	0	0.0035
R_H2O_transport_via_diffusion__extracellular_to_periplasm_	-37.2402	-37.2376	-37.2376
R_H2O_exchange	37.2402	37.2376	37.2376
R_L_aspartate_oxidase_001	0.0017	0	0
R_L_aspartate_oxidase_002	0	0	0.0017

Appendix A: Comparison of FBA solutions from different software tools

Reaction	*metano*	COBRA toolbox	SurreyFBA
R_L_aspartate_oxidase_003	0	0.0017	0
R_isopentenyl_diphosphate_D_isomerase	0	0.0002	-0.0016
R_1_hydroxy_2_methyl_2__E__butenyl_4_diphosphate_reductase__ipdp_	0.0016	0.0018	0
R_1_hydroxy_2_methyl_2__E__butenyl_4_diphosphate_reductase__dmpp_	0.0002	0	0.0018
R_ferroxidase	0.0013	0	0
R_oxygen_transport_via_diffusion__extracellular_to_periplasm_	16.2669	16.2656	16.2656
R_O2_exchange	-16.2669	-16.2656	-16.2656
R_fumarate_reductase_002	0	0.0005	0
R_Glycolate_oxidase_003	0	0.0005	0
R_Glycolate_oxidase_001	0.0005	0	0
R_Glycolate_oxidase_002	0	0	0.0005
R_polyphosphate_kinase_002	-2.2961	-2.2964	-2.2964
R_adenine_phosphoribosyltransferase	0	0.0003	0.0003
R_purine_nucleoside_phosphorylase__Adenosine_	-0.0003	0	0
R_adenosine_kinase	0.0003	0	0

Appendix B: FBA on *E. coli* model iJO1366 for the alternative scenario

FBA was performed for the alternative scenario described in Section 3.1.1.2. Like the original scenario, it describes a continuous growth environment with a constant glucose uptake rate of 11.0 mmol gDW^{-1} h^{-1}, and a forced acetate secretion rate of 6.4 mmol gDW^{-1} h^{-1}. Oxygen can be taken up from the environment at an arbitrary rate. The alternative scenario differs from the original scenario in a forced flux of (at least) 5 mmol gDW^{-1} h^{-1} through the reaction catalyzed by proton-translocating trans-hydrogenase (EC 1.6.1.2). The values for the glucose uptake rate, oxygen uptake rate, growth rate, acetate secretion rate, and CO_2 emission rate predicted by FBA under the two scenarios are shown in Table 22 along with experimentally determined values from a ^{13}C tracer analysis [154].

Table 22: Key flux values predicted by FBA for the original and alternative scenarios described in Section 3.1.1.2 compared to experimentally determined values. Experimental values were taken from [154]. The experimental values, which were obtained during growth in a bioreactor in continuous mode, are given with standard deviation from triplicate experiments. Growth rate is given in h^{-1}, all other fluxes in mmol gDW^{-1} h^{-1}.

	Experimental values	**FBA (original scenario)**	**FBA (alternative scenario)**
glucose uptake rate	11.0 ± 0.5	11.0 (fixed)	11.0 (fixed)
oxygen uptake rate	18.2 ± 0.8	16.5	16.7
growth rate	0.82 ± 0.02	0.85	0.85
CO_2 secretion rate	18.6 ± 0.5	18.2	18.4
acetate secretion rate	6.4 ± 1	6.4 (fixed)	6.4 (fixed)

At the given values for the glucose uptake rate and the acetate secretion rate, the FBA solutions for the two scenarios both accurately predict the CO_2 emission rate and the growth rate, while both yield somewhat too low values for the oxygen uptake rate. While FBA predicts nearly identical values for these key fluxes under the two scenarios, the predicted oxygen uptake rate is slightly higher (and thus closer to the experimentally determined value) in the alternative scenario.

References

1. Kell DB: **Systems biology, metabolic modelling and metabolomics in drug discovery and development**. *Drug Discov Today* 2006, **11**:1085–1092.

2. Jamshidi N, Palsson BØ: **Investigating the metabolic capabilities of Mycobacterium tuberculosis H37Rv using the in silico strain iNJ661 and proposing alternative drug targets**. *BMC Syst Biol* 2007, **1**:26.

3. Lee D-S, Burd H, Liu J, Almaas E, Wiest O, Barabási A-L, Oltvai ZN, Kapatral V: **Comparative genome-scale metabolic reconstruction and flux balance analysis of multiple Staphylococcus aureus genomes identify novel antimicrobial drug targets**. *J Bacteriol* 2009, **191**:4015–4024.

4. Huthmacher C, Hoppe A, Bulik S, Holzhütter H-G: **Antimalarial drug targets in Plasmodium falciparum predicted by stage-specific metabolic network analysis**. *BMC Syst Biol* 2010, **4**:120.

5. Folger O, Jerby L, Frezza C, Gottlieb E, Ruppin E, Shlomi T: **Predicting selective drug targets in cancer through metabolic networks**. *Mol Syst Biol* 2011, **7**:501.

6. Chang RL, Xie L, Xie L, Bourne PE, Palsson BØ: **Drug Off-Target Effects Predicted Using Structural Analysis in the Context of a Metabolic Network Model**. *PLoS Comput Biol* 2010, **6**:e1000938.

7. Patil KR, Åkesson M, Nielsen J: **Use of genome-scale microbial models for metabolic engineering**. *Curr Opin Biotechnol* 2004, **15**:64–69.

8. Hjersted JL, Henson MA: **Optimization of Fed-Batch Saccharomyces cerevisiae Fermentation Using Dynamic Flux Balance Models**. *Biotechnol Prog* 2006, **22**:1239–1248.

9. Khannapho C, Zhao H, Bonde BK, Kierzek AM, Avignone-Rossa CA, Bushell ME: **Selection of objective function in genome scale flux balance analysis for process feed development in antibiotic production**. *Metab Eng* 2008, **10**:227–233.

10. Puchałka J, Oberhardt MA, Godinho M, Bielecka A, Regenhardt D, Timmis KN, Papin JA, Martins dos Santos VAP: **Genome-Scale Reconstruction and Analysis of the Pseudomonas putida KT2440 Metabolic Network Facilitates Applications in Biotechnology**. *PLoS Comput Biol* 2008, **4**:e1000210.

11. Kim P-J, Lee D-Y, Kim TY, Lee KH, Jeong H, Lee SY, Park S: **Metabolite essentiality elucidates robustness of Escherichia coli metabolism**. *Proc Natl Acad Sci U S A* 2007, **104**:13638–13642.

12. Fong SS, Marciniak JY, Palsson BØ: **Description and interpretation of adaptive evolution of Escherichia coli K-12 MG1655 by using a genome-scale in silico metabolic model**. *J Bacteriol* 2003, **185**:6400–6408.

13. Price ND, Reed JL, Palsson BØ: **Genome-scale models of microbial cells: evaluating the consequences of constraints**. *Nat Rev Microbiol* 2004, **2**:886–897.

14. Becker SA, Feist AM, Mo ML, Hannum G, Palsson BØ, Herrgard MJ: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox**. *Nat Protoc* 2007, **2**:727–738.

15. Fell DA, Small JR: **Fat synthesis in adipose tissue. An examination of stoichiometric constraints**. *Biochem J* 1986, **238**:781–786.

16. Kauffman KJ, Prakash P, Edwards JS: **Advances in flux balance analysis**. *Curr Opin Biotechnol* 2003, **14**:491–496.

17. Orth JD, Thiele I, Palsson BØ: **What is flux balance analysis?** *Nat Biotechnol* 2010, **28**:245–248.

18. Kim TY, Sohn SB, Kim YB, Kim WJ, Lee SY: **Recent advances in reconstruction and applications of genome-scale metabolic models**. *Curr Opin Biotechnol* 2012, **23**:617–623.

19. Lakshmanan M, Koh G, Chung BKS, Lee D-Y: **Software applications for flux balance analysis**. *Brief Bioinform* 2012, doi:10.1242/dmm.004002.

20. Hoppe A, Hoffmann S, Gerasch A, Gille C, Holzhütter H-G: **FASIMU: flexible software for flux-balance computation series in large metabolic networks**. *BMC Bioinformatics* 2011, **12**:28.

21. Kelk SM, Olivier BG, Stougie L, Bruggeman FJ: **Optimal flux spaces of genome-scale stoichiometric models are determined by a few subnetworks**. *Sci Rep* 2012, **2**:580.

22. Noor E, Lewis NE, Milo R: **A proof for loop-law constraints in stoichiometric metabolic networks**. *BMC Syst Biol* 2012, **6**:140.

23. Oberhardt MA, Palsson BØ, Papin JA: **Applications of genome-scale metabolic reconstructions**. *Mol Syst Biol* 2009, **5**:320.

24. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BØ: **A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information**. *Mol Syst Biol* 2007, **3**:121.

25. Williams TCR, Poolman MG, Howden AJM, Schwarzlander M, Fell DA, Ratcliffe RG, Sweetlove LJ: **A genome-scale metabolic model accurately predicts fluxes in central carbon metabolism under stress conditions**. *Plant Physiol* 2010, **154**:311–323.

26. Yizhak K, Benyamini T, Liebermeister W, Ruppin E, Shlomi T: **Integrating quantitative proteomics and metabolomics with a genome-scale metabolic network model**. *Bioinformatics* 2010, **26**:i255–i260.

27. Herrgård MJ, Fong SS, Palsson BØ: **Identification of Genome-Scale Metabolic Network Models Using Experimentally Measured Flux Profiles**. *PLoS Comput Biol* 2006, **2**:e72.

28. Chechik G, Oh E, Rando O, Weissman J, Regev A, Koller D: **Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network**. *Nat Biotechnol* 2008, **26**:1251–1259.

29. Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng T-Y, Moody DB, Murray M, Galagan JE: **Interpreting Expression Data with Metabolic Flux Models: Predicting Mycobacterium tuberculosis Mycolic Acid Production**. *PLoS Comput Biol* 2009, **5**:e1000489.

30. Lewis NE, Nagarajan H, Palsson BØ: **Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods**. *Nat Rev Microbiol* 2012, **10**:291–305.

31. Kim HU, Kim TY, Lee SY: **Metabolic flux analysis and metabolic engineering of microorganisms**. *Mol BioSyst* 2008, **4**:113.

32. Ranganathan S, Suthers PF, Maranas CD: **OptForce: An Optimization Procedure for Identifying All Genetic Manipulations Leading to Targeted Overproductions**. *PLoS Comput Biol* 2010, **6**:e1000744.

33. Lee JW, Na D, Park JM, Lee J, Choi S, Lee SY: **Systems metabolic engineering of microorganisms for natural and non-natural chemicals**. *Nat Chem Biol* 2012, **8**:536–546.

34. Vongsangnak W, Figueiredo LF, Förster J, Weber T, Thykaer J, Stegmann E, Wohlleben W, Nielsen J: **Genome-scale metabolic representation of Amycolatopsis balhimycina**. *Biotechnol Bioeng* 2012, **109**:1798–1807.

35. Chung B, Lee D-Y: **Flux-sum analysis: a metabolite-centric approach for understanding the metabolic network**. *BMC Syst Biol* 2009, **3**:117.

36. Kim HU, Kim SY, Jeong H, Kim TY, Kim JJ, Choy HE, Yi KY, Rhee JH, Lee SY: **Integrative genome-scale metabolic analysis of Vibrio vulnificus for drug targeting and discovery**. *Mol Syst Biol* 2011, **7**:460.

37. Voit EO: **Design principles and operating principles: the yin and yang of optimal functioning**. *Math Biosci* 2003, **182**:81–92.

38. Koffas M, Stephanopoulos G: **Strain improvement by metabolic engineering: lysine production as a case study for systems biology**. *Curr Opin Biotechnol* 2005, **16**:361–366.

39. Huang D, Jia X, Wen J, Wang G, Yu G, Caiyin Q, Chen Y: **Metabolic Flux Analysis and Principal Nodes Identification for Daptomycin Production Improvement by Streptomyces roseosporus**. *Appl Biochem Biotech* 2011, **165**:1725–1739.

40. Hochachka PW: **Action of temperature on branch points in glucose and acetate metabolism**. *Comp Biochem Physiol* 1968, **25**:107–118.

41. Fell DA, Sauro HM: **Metabolic control and its analysis. Additional relationships between elasticities and control coefficients**. *Eur J Biochem* 1985, **148**:555–561.

42. Vallino JJ, Stephanopoulos G: **Metabolic flux distributions in Corynebacterium glutamicum during growth and lysine overproduction**. *Biotechnol Bioeng* 1993, **41**:633–646.

43. Klipp E, Herwig R, Kowald A, Wierling C, Lehrach H: *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH; 2005.

44. Papin JA, Palsson BO: **Topological analysis of mass-balanced signaling networks: a framework to obtain network properties including crosstalk**. *J Theor Biol* 2004, **227**:283–297.

45. Rual J-F, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, Li N, Berriz GF, Gibbons FD, Dreze M, Ayivi-Guedehoussou N, et al.: **Towards a proteome-scale map of the human protein–protein interaction network**. *Nature* 2005, **437**:1173–1178.

46. Sorek R, Cossart P: **Prokaryotic transcriptomics: a new view on regulation, physiology and pathogenicity**. *Nat Rev Genet* 2009, **11**:9–16.

47. Carey M, Peterson CL, Smale ST: *Transcriptional Regulation in Eukaryotes: Concepts, Strategies, and Techniques*. 2nd ed. Cold Spring Harbor Laboratory Press; 2008.

48. Likić VA, McConville MJ, Lithgow T, Bacic A: **Systems Biology: The Next Frontier for Bioinformatics**. *Adv Bioinformatics* 2010, **2010**:268925.

49. Kitano H: **Systems Biology: A Brief Overview**. *Science* 2002, **295**:1662–1664.

50. Butcher EC, Berg EL, Kunkel EJ: **Systems biology in drug discovery**. *Nat Biotechnol* 2004, **22**:1253–1259.

51. Molina F, Dehmer M, Perco P, Graber A, Girolami M, Spasovski G, Schanstra JP, Vlahou A: **Systems biology: opening new avenues in clinical research**. *Nephrol Dial Transplant* 2010, **25**:1015–1018.

52. Shapiro BE, Levchenko A, Mjolsness E: **Automatic Model Generation for Signal Transduction with Applications to MAP-Kinase Pathways**. In *Foundations of Systems Biology*. Edited by Kitano H. MIT Press; 2001.

53. Flöttmann M, Schaber J, Hoops S, Klipp E, Mendes P: **ModelMage: a tool for automatic model generation, selection and management**. *Genome Inform* 2008, **20**:52–63.

54. Henry CS, DeJongh M, Best AA, Frybarger PM, Linsay B, Stevens RL: **High-throughput generation, optimization and analysis of genome-scale metabolic models**. *Nat Biotechnol* 2010, **28**:977–982.

55. Madigan MT, Martinko JM: *Brock Biology of Microorganisms*. 11th ed. Pearson Prentice Hall; 2006.

56. Gaffney EA, Monk NAM: **Gene Expression Time Delays and Turing Pattern Formation Systems**. *Bull Math Biol* 2006, **68**:99–130.

57. Hargrove JL: **Microcomputer-assisted kinetic modeling of mammalian gene expression**. *FASEB J* 1993, **7**:1163–1170.

58. Wehrman T, Kleaveland B, Her J-H, Balint RF, Blau HM: **Protein-protein interactions monitored in mammalian cells via complementation of beta-lactamase enzyme fragments**. *Proc Natl Acad Sci U S A* 2002, **99**:3469–3474.

59. Steuer R: **Review: On the analysis and interpretation of correlations in metabolomic data**. *Brief Bioinform* 2006, **7**:151–158.

60. Sauer U: **High-throughput phenomics: experimental methods for mapping fluxomes**. *Curr Opin Biotechnol* 2004, **15**:58–63.

61. Deville Y, Gilbert D, van Helden J, Wodak SJ: **An overview of data models for the analysis of biochemical pathways**. *Brief Bioinform* 2003, **4**:246–259.

62. Ehrentreich F, Schomburg D: **Dynamic generation and qualitative analysis of metabolic pathways by a joint database/graph theoretical approach.** *Funct Integr Genomics* 2003, **3**:189–196.

63. Borenstein E, Kupiec M, Feldman MW, Ruppin E: **Large-scale reconstruction and phylogenetic analysis of metabolic environments**. *Proc Natl Acad Sci U S A* 2008, **105**:14482–14487.

64. Aittokallio T: **Graph-based methods for analysing networks in cell biology**. *Brief Bioinform* 2006, **7**:243–255.

65. Ma H-W, Zhao X-M, Yuan Y-J, Zeng A-P: **Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph**. *Bioinformatics* 2004, **20**:1870–1876.

66. Rahman SA, Schomburg D: **Observing local and global properties of metabolic pathways: "load points" and "choke points" in the metabolic networks**. *Bioinformatics* 2006, **22**:1767–1774.

67. Klipp E: **Modelling dynamic processes in yeast**. *Yeast* 2007, **24**:943–959.

68. Steuer R: **Computational approaches to the topology, stability and dynamics of metabolic networks**. *Phytochemistry* 2007, **68**:2139–2151.

69. Teusink B, Passarge J, Reijenga CA, Esgalhado E, van der Weijden CC, Schepper M, Walsh MC, Bakker BM, van Dam K, Westerhoff HV, et al.: **Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry**. *Eur J Biochem* 2000, **267**:5313–5329.

70. Gombert AK, Nielsen J: **Mathematical modelling of metabolism**. *Curr Opin Biotechnol* 2000, **11**:180–186.

71. Machado D, Costa RS, Ferreira EC, Rocha I, Tidor B: **Exploring the gap between dynamic and constraint-based models of metabolism**. *Metab Eng* 2012, **14**:112–119.

72. Zi Z, Feng Z, Chapnick DA, Dahl M, Deng D, Klipp E, Moustakas A, Liu X: **Quantitative analysis of transient and sustained transforming growth factor-β signaling dynamics**. *Mol Syst Biol* 2011, **7**:492.

73. Becker V, Timmer J, Klingmüller U: **Receptor Dynamics in Signaling**. In *Advances in Systems Biology*. Edited by Goryanin II, Goryachev AB. Springer New York; 2012:313–323.

74. Klipp E, Nordlander B, Krüger R, Gennemark P, Hohmann S: **Integrative model of the response of yeast to osmotic shock**. *Nat Biotechnol* 2005, **23**:975–982.

75. Covert MW, Xiao N, Chen TJ, Karr JR: **Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli**. *Bioinformatics* 2008, **24**:2044–2050.

76. Feist AM, Palsson BØ: **The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli**. *Nat Biotechnol* 2008, **26**:659–667.

77. Reed JL, Vo TD, Schilling CH, Palsson BO: **An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR)**. *Genome Biol* 2003, **4**:R54.

78. Shinfuku Y, Sorpitiporn N, Sono M, Furusawa C, Hirasawa T, Shimizu H: **Development and experimental verification of a genome-scale metabolic model for Corynebacterium glutamicum**. *Microb Cell Fact* 2009, **8**:43.

79. Oberhardt MA, Puchałka J, Fryer KE, Martins dos Santos VAP, Papin JA: **Genome-scale metabolic network analysis of the opportunistic pathogen Pseudomonas aeruginosa PAO1**. *J Bacteriol* 2008, **190**:2790–2803.

80. Srivastava R: **Genome-Scale Analysis of Metabolic Networks**. In *Systems Analysis of Biological Networks*. Edited by Jayaraman A, Hahn J. Artech House; 2009:95–110.

81. Palsson BØ: *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press; 2006.

82. Burton AC: **The properties of the steady state compared to those of equilibrium as shown in characteristic biological behavior**. *J Cell Comp Physiol* 1939, **14**:327–349.

83. Voit EO: *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press; 2000.

84. Klamt S, Schuster S, Gilles ED: **Calculability analysis in underdetermined metabolic networks illustrated by a model of the central metabolism in purple nonsulfur bacteria**. *Biotechnol Bioeng* 2002, **77**:734–751.

85. Durot M, Bourguignon P-Y, Schachter V: **Genome-scale models of bacterial metabolism: reconstruction and applications**. *FEMS Microbiol Rev* 2009, **33**:164–190.

86. Navid A, Almaas E: **Genome-scale reconstruction of the metabolic network in Yersinia pestis, strain 91001**. *Mol BioSyst* 2009, **5**:368–375.

87. Hartwig I, Schomburg D: **Analysis of a genome-scale metabolic model of Yersinia pseudotuberculosis**. Manuscript in preparation.

88. Perumal D, Samal A, Sakharkar KR, Sakharkar MK: **Targeting multiple targets in Pseudomonas aeruginosa PAO1 using flux balance analysis of a reconstructed genome-scale metabolic network**. *J Drug Target* 2011, **19**:1–13.

89. Kjeldsen KR, Nielsen J: **In silico genome-scale reconstruction and validation of the Corynebacterium glutamicum metabolic network**. *Biotechnol Bioeng* 2009, **102**:583–597.

90. Bekaert M, Conant GC: **Copy Number Alterations among Mammalian Enzymes Cluster in the Metabolic Network**. *Mol Biol Evol* 2010, **28**:1111–1121.

91. Ederer M, Schlatter R, Witt J, Feuer R, Bona-Lovasz J, Henkel S, Sawodny O: **An introduction to kinetic, constraint-based and Boolean modeling in systems biology**. In *Control Applications (CCA), 2010 IEEE International Conference on*. 2010:129–134.

92. Llaneras F, Sala A, Picó J: **A possibilistic framework for constraint-based metabolic flux analysis**. *BMC Syst Biol* 2009, **3**:79.

93. Thiele I, Palsson BØ: **A protocol for generating a high-quality genome-scale metabolic reconstruction**. *Nat Protoc* 2010, **5**:93–121.

94. Varma A, Boesch BW, Palsson BO: **Stoichiometric interpretation of Escherichia coli glucose catabolism under various oxygenation rates**. *Appl Environ Microbiol* 1993, **59**:2465–2473.

95. Price ND, Famili I, Beard DA, Palsson BØ: **Extreme pathways and Kirchhoff's second law**. *Biophys J* 2002, **83**:2879–2882.

96. Nigam R, Liang S: **A pivoting algorithm for metabolic networks in the presence of thermodynamic constraints**. In *Proc IEEE Comput Syst Bioinform Conf*. 2005:259–267.

97. Masterton WL, Hurley CN: *Chemistry : Principles and Reactions*. 6th ed. Brooks/Cole, Cengage Learning; 2008.

98. Poolman MG, Miguet L, Sweetlove LJ, Fell DA: **A genome-scale metabolic model of Arabidopsis and some of its properties**. *Plant Physiol* 2009, **151**:1570–1581.

99. Ponce de León M, Cancela H, Acerenza L: **A Strategy to Calculate the Patterns of Nutrient Consumption by Microorganisms Applying a Two-Level Optimisation Principle to Reconstructed Metabolic Networks**. *J Biol Phys* 2008, **34**:73–90.

100. Price ND, Thiele I, Palsson BØ: **Candidate States of Helicobacter pylori's Genome-Scale Metabolic Network upon Application of "Loop Law" Thermodynamic Constraints**. *Biophys J* 2006, **90**:3919–3928.

101. Kümmel A, Panke S, Heinemann M: **Systematic assignment of thermodynamic constraints in metabolic network models**. *BMC Bioinformatics* 2006, **7**:512.

102. Wright J, Wagner A: **Exhaustive identification of steady state cycles in large stoichiometric networks**. *BMC Syst Biol* 2008, **2**:61.

103. Schellenberger J, Lewis NE, Palsson BØ: **Elimination of Thermodynamically Infeasible Loops in Steady-State Metabolic Models**. *Biophys J* 2011, **100**:544–553.

104. Feist AM, Herrgård MJ, Thiele I, Reed JL, Palsson BØ: **Reconstruction of biochemical networks in microorganisms**. *Nat Rev Microbiol* 2008, **7**:129–143.

105. Quester S, Schomburg D: **EnzymeDetector: an integrated enzyme function prediction tool and database**. *BMC Bioinformatics* 2011, **12**:376.

106. Scheer M, Grote A, Chang A, Schomburg I, Munaretto C, Rother M, Söhngen C, Stelzer M, Thiele J, Schomburg D: **BRENDA, the enzyme information system in 2011**. *Nucleic Acids Res* 2011, **39**:D670–676.

107. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M: **KEGG for integration and interpretation of large-scale molecular data sets**. *Nucleic Acids Res* 2011, **40**:D109–D114.

108. The UniProt Consortium: **Ongoing and future developments at the Universal Protein Resource**. *Nucleic Acids Res* 2010, **39**:D214–D219.

109. Bannert C, Welfle A, aus dem Spring C, Schomburg D: **BrEPS: a flexible and automatic protocol to compute enzyme-specific sequence profiles for functional annotation**. *BMC Bioinformatics* 2010, **11**:589.

110. Börner J, Buchinger S, Schomburg D: **A high-throughput method for microbial metabolome analysis using gas chromatography/mass spectrometry**. *Anal Biochem* 2007, **367**:143–151.

111. Poblete-Castro I, Escapa IF, Jäger C, Puchalka J, Lam CMC, Schomburg D, Prieto MA, Martins dos Santos VAP: **The metabolic response of P. putida KT2442 producing high levels of polyhydroxyalkanoate under single- and multiple-nutrient-limited growth: highlights from a multi-level omics approach**. *Microb Cell Fact* 2012, **11**:34.

112. Bennett BD, Kimball EH, Gao M, Osterhout R, Van Dien SJ, Rabinowitz JD: **Absolute metabolite concentrations and implied enzyme active site occupancy in Escherichia coli**. *Nat Chem Biol* 2009, **5**:593–599.

113. Zamboni N, Kümmel A, Heinemann M: **anNET: a tool for network-embedded thermodynamic analysis of quantitative metabolome data**. *BMC Bioinformatics* 2008, **9**:199.

114. Wittmann C: **Metabolic Flux Analysis Using Mass Spectrometry**. In *Tools and Applications of Biochemical Engineering Science*. Edited by Schügerl K, Zeng A-P. Springer-Verlag; 2002:39–64.

115. Wolak J, Rahimi-Keshari K, Jeffries RE, Joy MP, Todd A, Pediatitakis P, Dewar BJ, Winnike JH, Favorov O, Elston TC, et al.: **Noninvasive Fluxomics in Mammals by Nuclear Magnetic Resonance Spectroscopy**. In *The Handbook of Metabolomics*. Edited by Fan TW-M, Lane AN, Higashi RM. Humana Press; 2012:321–392.

116. Wiechert W: **13C Metabolic Flux Analysis**. *Metab Eng* 2001, **3**:195–206.

117. Sauer U: **Metabolic networks in motion: 13C-based flux analysis**. *Mol Syst Biol* 2006, **2**:62.

118. Metallo CM, Walther JL, Stephanopoulos G: **Evaluation of 13C isotopic tracers for metabolic flux analysis in mammalian cells**. *J Biotechnol* 2009, **144**:167–174.

119. Wittmann C: **Fluxome analysis using GC-MS**. *Microb Cell Fact* 2007, **6**:6.

120. Hiller K, Metallo C, Stephanopoulos G: **Elucidation of cellular metabolism via metabolomics and stable-isotope assisted metabolomics**. *Curr Pharm Biotechnol* 2011, **12**:1075–1086.

121. Maier K, Hofmann U, Reuss M, Mauch K: **Identification of metabolic fluxes in hepatic cells from transient 13C-labeling experiments: Part II. Flux estimation**. *Biotechnol Bioeng* 2008, **100**:355–370.

122. Bochner BR, Gadzinski P, Panomitros E: **Phenotype microarrays for high-throughput phenotypic testing and assay of gene function**. *Genome Res* 2001, **11**:1246–1255.

123. Bochner BR: **Global phenotypic characterization of bacteria**. *FEMS Microbiol Rev* 2009, **33**:191–205.

124. Schlegel HG: *Allgemeine Mikrobiologie*. 6th ed. Thieme; 1985.

125. Banks RB: *Growth and Diffusion Phenomena: Mathematical Frameworks and Applications*. Springer-Verlag; 1993.

126. Kaper JB, Nataro JP, Mobley HLT: **Pathogenic Escherichia coli**. *Nat Rev Microbiol* 2004, **2**:123–140.

127. Racker E: **Enzymatic formation and breakdown of pentose phosphate**. *Fed Proc* 1948, **7**:180.

128. Rohmer M, Knani M, Simonin P, Sutter B, Sahm H: **Isoprenoid biosynthesis in bacteria: a novel pathway for the early steps leading to isopentenyl diphosphate**. *Biochem J* 1993, **295 (Pt 2)**:517–524.

129. Bausch C, Peekhaus N, Utz C, Blais T, Murray E, Lowary T, Conway T: **Sequence analysis of the GntII (subsidiary) system for gluconate metabolism reveals a novel pathway for L-idonic acid catabolism in Escherichia coli**. *J Bacteriol* 1998, **180**:3704–3710.

130. Jin RZ, Tang JC-T, Lin ECC: **Experimental evolution of a novel pathway for glycerol dissimilation in Escherichia coli**. *J Mol Evol* 1983, **19**:429–436.

131. Peretti SW, Bailey JE: **Simulations of host-plasmid interactions in Escherichia coli: Copy number, promoter strength, and ribosome binding site strength effects on metabolic activity and plasmid gene expression**. *Biotechnol Bioeng* 1987, **29**:316–328.

132. Palsson BO, Joshi A: **On the dynamic order of structured Escherichia coli growth models**. *Biotechnol Bioeng* 1987, **29**:789–792.

133. Neijssel OM, Tempest DW: **The regulation of carbohydrate metabolism in Klebsiella aerogenes NCTC 418 organisms, growing in chemostat culture**. *Arch Microbiol* 1975, **106**:251–258.

134. Vemuri GN, Altman E, Sangurdekar DP, Khodursky AB, Eiteman MA: **Overflow Metabolism in Escherichia coli during Steady-State Growth: Transcriptional Regulation and Effect of the Redox Ratio**. *Appl Environ Microbiol* 2006, **72**:3653–3661.

135. Blattner FR, Plunkett G III, Bloch CA, Perna NT, Burland V, Riley M, Collado-Vides J, Glasner JD, Rode CK, Mayhew GF, et al.: **The complete genome sequence of Escherichia coli K-12**. *Science* 1997, **277**:1453–1462.

136. Zhaxybayeva O, Swithers KS, Lapierre P, Fournier GP, Bickhart DM, DeBoy RT, Nelson KE, Nesbo CL, Doolittle WF, Gogarten JP, et al.: **On the chimeric nature, thermophilic origin, and phylogenetic placement of the Thermotogales**. *Proc Natl Acad Sci U S A* 2009, **106**:5865–5870.

137. Gupta RS: **Protein phylogenies and signature sequences: A reappraisal of evolutionary relationships among archaebacteria, eubacteria, and eukaryotes**. *Microbiol Mol Biol Rev* 1998, **62**:1435–1491.

138. Schröder C, Selig M, Schönheit P: **Glucose fermentation to acetate, CO2 and H2 in the anaerobic hyperthermophilic eubacterium Thermotoga maritima: involvement of the Embden-Meyerhof pathway**. *Arch Microbiol* 1994, **161**:460–470.

139. Adams MWW: **Biochemical diversity among sulfur-dependent, hyperthermophilic microorganisms**. *FEMS Microbiol Rev* 1994, **15**:261–277.

140. Hedges SB: **The origin and evolution of model organisms**. *Nat Rev Genet* 2002, **3**:838–849.

141. Costa MS, Santos H, Galinski EA: **An overview of the role and diversity of compatible solutes in Bacteria and Archaea**. In *Biotechnology of Extremophiles*. Edited by Antranikian G. Springer Berlin Heidelberg; 1998:117–153.

142. Huber R, Langworthy TA, König H, Thomm M, Woese CR, Sleytr UB, Stetter KO: **Thermotoga maritima sp. nov. represents a new genus of unique extremely thermophilic eubacteria growing up to 90°C**. *Arch Microbiol* 1986, **144**:324–333.

143. Nelson KE, Clayton RA, Gill SR, Gwinn ML, Dodson RJ, Haft DH, Hickey EK, Peterson JD, Nelson WC, Ketchum KA, et al.: **Evidence for lateral gene transfer between Archaea and bacteria from genome sequence of Thermotoga maritima**. *Nature* 1999, **399**:323–329.

144. Hernandez D, Dias FM, Rowe JJ: **Nitrate transport and its regulation by O2 in Pseudomonas aeruginosa**. *Arch Biochem Biophys* 1991, **286**:159–163.

145. Høiby N, Johansen HK, Moser C, Ciofu O: **Clinical Relevance of Pseudomonas aeruginosa: A Master of Adaptation and Survival Strategies**. In *Pseudomonas: Model Organism, Pathogen, Cell Factory*. Edited by Rehm BHA. Wiley-VCH; 2008:25–44.

146. Soberón-Chávez G, Lépine F, Déziel E: **Production of rhamnolipids by Pseudomonas aeruginosa**. *Appl Microbiol Biotechnol* 2005, **68**:718–725.

147. Holloway BW: **Genetic Recombination in Pseudomonas aeruginosa**. *J Gen Microbiol* 1955, **13**:572–581.

148. Stover CK, Pham XQ, Erwin AL, Mizoguchi SD, Warrener P, Hickey MJ, Brinkman FS, Hufnagle WO, Kowalik DJ, Lagrou M, et al.: **Complete genome sequence of Pseudomonas aeruginosa PAO1, an opportunistic pathogen**. *Nature* 2000, **406**:959–964.

149. Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, Palsson BØ: **A comprehensive genome-scale reconstruction of Escherichia coli metabolism —2011**. *Mol Syst Biol* 2011, **7**:535.

150. Riley M, Abe T, Arnaud MB, Berlyn MKB, Blattner FR, Chaudhuri RR, Glasner JD, Horiuchi T, Keseler IM, Kosuge T, et al.: **Escherichia coli K-12: a cooperatively developed annotation snapshot--2005**. *Nucleic Acids Res* 2006, **34**:1–9.

151. Keseler IM, Collado-Vides J, Gama-Castro S, Ingraham J, Paley S, Paulsen IT, Peralta-Gil M, Karp PD: **EcoCyc: a comprehensive database resource for Escherichia coli**. *Nucleic Acids Res* 2005, **33**:D334–337.

152. Henry CS, Jankowski MD, Broadbelt LJ, Hatzimanikatis V: **Genome-Scale Thermodynamic Analysis of Escherichia coli Metabolism**. *Biophys J* 2006, **90**:1453–1461.

153. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, et al.: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models**. *Bioinformatics* 2003, **19**:524–531.

154. Fischer E, Zamboni N, Sauer U: **High-throughput metabolic flux analysis based on gas chromatography-mass spectrometry derived 13C constraints**. *Anal Biochem* 2004, **325**:308–316.

155. Zhang Y, Thiele I, Weekes D, Li Z, Jaroszewski L, Ginalski K, Deacon AM, Wooley J, Lesley SA, Wilson IA, et al.: **Three-Dimensional Structural View of the Central Metabolic Network of Thermotoga maritima**. *Science* 2009, **325**:1544–1549.

156. Feng L, Tumbula-Hansen D, Min B, Namgoog S, Salazar J, Orellana O, Söll D: **Transfer RNA-Dependent Amidotransferases: Key Enzymes for Asn-tRNA and Gln-tRNA Synthesis in Nature**. *Eurekah Bioscience* 2005, **1**:86–91.

157. Chvátal V: *Linear Programming*. W.H. Freeman; 1983.

158. Schuetz R, Kuepfer L, Sauer U: **Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli**. *Mol Syst Biol* 2007, **3**:119.

159. Rolfsson O, Palsson BØ, Thiele I: **The human metabolic reconstruction Recon 1 directs hypotheses of novel human metabolic functions**. *BMC Syst Biol* 2011, **5**:155.

160. Samal A, Wagner A, Martin OC: **Environmental versatility promotes modularity in genome-scale metabolic networks**. *BMC Syst Biol* 2011, **5**:135.

161. Lee D-Y, Fan LT, Park S, Lee SY, Shafie S, Bertók B, Friedler F: **Complementary identification of multiple flux distributions and multiple metabolic pathways**. *Metab Eng* 2005, **7**:182–200.

162. Mahadevan R, Schilling CH: **The effects of alternate optimal solutions in constraint-based genome-scale metabolic models**. *Metab Eng* 2003, **5**:264–276.

163. Gudmundsson S, Thiele I: **Computationally efficient flux variability analysis**. *BMC Bioinformatics* 2010, **11**:489.

164. Fischer G, Neuvéglise C, Durrens P, Gaillardin C, Dujon B: **Evolution of gene order in the genomes of two related yeast species**. *Genome Res* 2001, **11**:2009–2019.

165. Lynch M, Conery JS: **The Evolutionary Fate and Consequences of Duplicate Genes**. *Science* 2000, **290**:1151–1155.

166. Segrè D, Vitkup D, Church GM: **Analysis of optimality in natural and perturbed metabolic networks**. *Proc Natl Acad Sci U S A* 2002, **99**:15112–15117.

167. Kim J, Reed JL, Maravelias CT: **Large-Scale Bi-Level Strain Design Approaches and Mixed-Integer Programming Solution Techniques**. *PLoS ONE* 2011, **6**:e24162.

168. Shlomi T, Berkman O, Ruppin E: **Regulatory on/off minimization of metabolic flux changes after genetic perturbations**. *Proc Natl Acad Sci U S A* 2005, **102**:7695–7700.

169. Tang YJ, Martin HG, Myers S, Rodriguez S, Baidoo EEK, Keasling JD: **Advances in analysis of microbial metabolic fluxes via 13C isotopic labeling**. *Mass Spectrom Rev* 2009, **28**:362–375.

170. Riemer SA, Rex R, Schomburg D: **A metabolite-centric view on flux distributions in genome-scale metabolic models**. *BMC Syst Biol* 2013, **7**:33.

171. Gerhart SL: **Assertions and APL programming**. In *Proceedings of Seventh International Conference on APL*. Edited by Abrams PS, Haegi HR, Trumpy S. ACM; 1975:138–147.

172. Kotz JC, Treichel PM, Townsend JR: *Chemistry and Chemical Reactivity*. 7th ed. Thomson Brooks/Cole; 2009.

173. Alberty RA: *Biochemical Thermodynamics: Applications of Mathematica*. Wiley-Interscience; 2006.

174. Goldberg RN, Tewari YB, Bhat TN: **Thermodynamics of enzyme-catalyzed reactions—a database for quantitative biochemistry**. *Bioinformatics* 2004, **20**:2874–2877.

175. Mavrovouniotis ML: **Group contributions for estimating standard Gibbs energies of formation of biochemical compounds in aqueous solution**. *Biotechnol Bioeng* 1990, **36**:1070–1082.

176. Alberty RA: *Thermodynamics of Biochemical Reactions*. John Wiley & Sons; 2003.

177. Brazma A, Krestyaninova M, Sarkans U: **Standards for systems biology**. *Nat Rev Genet* 2006, **7**:593–605.

178. Gropp W, Lusk E, Skjellum A: *Using MPI : Portable Parallel Programming with the Message-Passing Interface*. 2nd ed. MIT Press; 1999.

179. Birgin EG, Martínez JM: **Structured minimal-memory inexact quasi-Newton method and secant preconditioners for augmented Lagrangian optimization**. *Comput Optim Appl* 2007, **39**:1–16.

180. Meyer B: *Object-Oriented Software Construction*. 2nd ed. Prentice-Hall; 1997.

181. Schellenberger J, Park JO, Conrad TM, Palsson BØ: **BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions**. *BMC Bioinformatics* 2010, **11**:213.

182. Xu Z, Sun X, Yu S: **Genome-scale analysis to the impact of gene deletion on the metabolism of E. coli: constraint-based simulation approach**. *BMC Bioinformatics* 2009, **10**:S62.

183. Nair NU, Goyal N, Chandra NR: **Enhanced flux balance analysis to model metabolic networks**. In *BCB '10 Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*. ACM Press; 2010:358–361.

184. Ahuja RK, Magnanti TL, Orlin JB: *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall; 1993.

185. Gevorgyan A, Bushell ME, Avignone-Rossa C, Kierzek AM: **SurreyFBA: a command line tool and graphics user interface for constraint-based modeling of genome-scale metabolic reaction networks**. *Bioinformatics* 2011, **27**:433–434.

186. Schellenberger J, Que R, Fleming RMT, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, et al.: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0**. *Nat Protoc* 2011, **6**:1290–1307.

187. Rex R, Bill N, Schmidt-Hohagen K, Schomburg D: **Swimming in light: a large-scale computational analysis of the metabolism of Dinoroseobacter shibae**. submitted for publication.

188. Tanenbaum AS: *Distributed Operating Systems*. Prentice Hall; 1995.

189. Friedl JEF: *Mastering Regular Expressions*. 3rd ed. O'Reilly Media; 2006.

190. Sigurdsson MI, Jamshidi N, Steingrimsson E, Thiele I, Palsson BØ: **A detailed genome-wide reconstruction of mouse metabolism based on human Recon 1**. *BMC Syst Biol* 2010, **4**:140.

191. Suthers PF, Zomorrodi A, Maranas CD: **Genome-scale gene/reaction essentiality and synthetic lethality analysis**. *Mol Syst Biol* 2009, **5**:301.

192. Chavali AK, D'Auria KM, Hewlett EL, Pearson RD, Papin JA: **A metabolic network approach for the identification and prioritization of antimicrobial drug targets**. *Trends Microbiol* 2012, **20**:113–123.

193. Lee SJ, Lee D-Y, Kim TY, Kim BH, Lee J, Lee SY: **Metabolic Engineering of Escherichia coli for Enhanced Production of Succinic Acid, Based on Genome Comparison and In Silico Gene Knockout Simulation**. *Appl Environ Microbiol* 2005, **71**:7880–7887.

194. Andersen MR, Nielsen ML, Nielsen J: **Metabolic model integration of the bibliome, genome, metabolome and reactome of Aspergillus niger**. *Mol Syst Biol* 2008, **4**:178.

195. Becker SA, Palsson BØ: **Genome-scale reconstruction of the metabolic network in Staphylococcus aureus N315: an initial draft to the two-dimensional annotation**. *BMC Microbiol* 2005, **5**:8.

196. Raghunathan A, Price ND, Galperin MY, Makarova KS, Purvine S, Picone AF, Cherny T, Xie T, Reilly TJ, Munson R, et al.: **In Silico Metabolic Model and Protein Expression of Haemophilus influenzae Strain Rd KW20 in Rich Medium**. *OMICS* 2004, **8**:25–41.

197. Ulas T, Riemer SA, Zaparty M, Siebers B, Schomburg D: **Genome-Scale Reconstruction and Analysis of the Metabolic Network in the Hyperthermophilic Archaeon Sulfolobus solfataricus**. *PLoS ONE* 2012, **7**:e43401.

198. Blank LM, Ebert BE, Buehler K, Bühler B: **Redox Biocatalysis and Metabolism: Molecular Mechanisms and Metabolic Network Analysis**. *Antioxid Redox Signal* 2010, **13**:349–394.

199. Greie J-C, Deckers-Hebestreit G, Altendorf K: **Energy-Transducing Ion Pumps in Bacteria: Structure and Function of ATP Synthases**. In *Microbial Transport Systems*. Edited by Winkelmann G. Wiley-VCH; 2001:23–45.

200. Michal G, Schomburg D: *Biochemical Pathways: An Atlas of Biochemistry and Molecular Biology*. 2nd ed. John Wiley & Sons; 2012.

201. Chulavatnatol M, Atkinson DE: **Phosphoenolpyruvate synthetase from Escherichia coli. Effects of adenylate energy charge and modifier concentrations**. *J Biol Chem* 1973, **248**:2712–2715.

202. Curtis SJ, Epstein W: **Phosphorylation of D-glucose in Escherichia coli mutants defective in glucosephosphotransferase, mannosephosphotransferase, and glucokinase**. *J Bacteriol* 1975, **122**:1189–1199.

203. Paulsen IT, Reizer J, Jin RZ, Lin EC, Saier MH Jr: **Functional genomic studies of dihydroxyacetone utilization in Escherichia coli**. *Microbiology (Reading, Engl)* 2000, **146 (Pt 10)**:2343–2344.

204. Erni B, Siebold C, Christen S, Srinivas A, Oberholzer A, Baumann U: **Small substrate, big surprise: fold, function and phylogeny of dihydroxyacetone kinases**. *Cell Mol Life Sci* 2006, **63**:890–900.

205. Subedi KP, Kim I, Kim J, Min B, Park C: **Role of GldA in dihydroxyacetone and methylglyoxal metabolism of Escherichia coli K12**. *FEMS Microbiol Lett* 2008, **279**:180–187.

206. Schürmann M, Sprenger GA: **Fructose-6-phosphate aldolase is a novel class I aldolase from Escherichia coli and is related to a novel group of bacterial transaldolases**. *J Biol Chem* 2001, **276**:11055–11061.

207. Gunsalus IC, Horecker BL, Wood WA: **Pathways of carbohydrate metabolism in microorganisms**. *Bacteriol Rev* 1955, **19**:79–128.

208. Sauer U, Canonaco F, Heri S, Perrenoud A, Fischer E: **The Soluble and Membrane-bound Transhydrogenases UdhA and PntAB Have Divergent Functions in NADPH Metabolism of Escherichia coli**. *J Biol Chem* 2003, **279**:6613–6619.

209. Arnér ESJ, Holmgren A: **Physiological functions of thioredoxin and thioredoxin reductase**. *Eur J Biochem* 2000, **267**:6102–6109.

210. Meyer Y, Buchanan BB, Vignols F, Reichheld J-P: **Thioredoxins and glutaredoxins: unifying elements in redox biology**. *Annu Rev Genet* 2009, **43**:335–367.

211. Carmel-Harel O, Storz G: **Roles of the glutathione- and thioredoxin-dependent reduction systems in the Escherichia coli and Saccharomyces cerevisiae responses to oxidative stress**. *Annu Rev Microbiol* 2000, **54**:439–461.

212. Klipp E, Liebermeister W: **Mathematical modeling of intracellular signaling pathways**. *BMC Neurosci* 2006, **7**:S10.

213. Sigüenza R, Flores N, Hernández G, Martínez A, Bolivar F, Valle F: **Kinetic characterization in batch and continuous culture of Escherichia coli mutants affected in phosphoenolpyruvate metabolism: differences in acetic acid production**. *World J Microbiol Biotechnol* 1999, **15**:587–592.

214. Nakahigashi K, Toya Y, Ishii N, Soga T, Hasegawa M, Watanabe H, Takai Y, Honma M, Mori H, Tomita M: **Systematic phenome analysis of Escherichia coli multiple-knockout mutants reveals hidden reactions in central carbon metabolism**. *Mol Syst Biol* 2009, **5**:306.

215. Guerinot ML: **Microbial Iron Transport**. *Annu Rev Microbiol* 1994, **48**:743–772.

216. Conners SB, Montero CI, Comfort DA, Shockley KR, Johnson MR, Chhabra SR, Kelly RM: **An Expression-Driven Approach to the Prediction of Carbohydrate Transport and Utilization Regulons in the Hyperthermophilic Bacterium Thermotoga maritima**. *J Bacteriol* 2005, **187**:7267–7282.

217. Barabote RD, Saier MH: **Comparative Genomic Analyses of the Bacterial Phosphotransferase System**. *Microbiol Mol Biol Rev* 2005, **69**:608–634.

218. Galperin MY, Noll KM, Romano AH: **The glucose transport system of the hyperthermophilic anaerobic bacterium Thermotoga neapolitana**. *Appl Environ Microbiol* 1996, **62**:2915–2918.

219. Schut GJ, Adams MW: **The iron-hydrogenase of Thermotoga maritima utilizes ferredoxin and NADH synergistically: a new perspective on anaerobic hydrogen production**. *J Bacteriol* 2009, **191**:4451–4457.

220. Liu J-Q, Dairi T, Itoh N, Kataoka M, Shimizu S, Yamada H: **Diversity of microbial threonine aldolases and their application**. *J Mol Catal B Enzym* 2000, **10**:107–115.

221. Kielkopf CL, Burley SK: **X-ray structures of threonine aldolase complexes: structural basis of substrate recognition**. *Biochemistry* 2002, **41**:11711–11720.

222. Snell K: **The duality of pathways for serine biosynthesis is a fallacy**. *Trends Biochem Sci* 1986, **11**:241–243.

223. Lang M, Stelzer M, Schomburg D: **BKM-react, an integrated biochemical reaction database**. *BMC Biochem* 2011, **12**:42.

224. Caspi R, Altman T, Dreher K, Fulcher CA, Subhraveti P, Keseler IM, Kothari A, Krummenacker M, Latendresse M, Mueller LA, et al.: **The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases**. *Nucleic Acids Res* 2011, **40**:D742–D753.

225. Lang M: **gapFiller – Entwicklung eines Tools zur Identifizierung fehlender Enzyme in biochemischen Pathways**. Doctoral dissertation, Technische Universität Braunschweig, 2012.

226. Latendresse M, Krummenacker M, Trupp M, Karp PD: **Construction and completion of flux balance models from pathway databases**. *Bioinformatics* 2012, **28**:388–396.

227. Fischer CR, Klein-Marcuschamer D, Stephanopoulos G: **Selection and optimization of microbial hosts for biofuels production**. *Metab Eng* 2008, **10**:295–304.

228. Chotani G, Dodge T, Hsu A, Kumar M, LaDuca R, Trimbur D, Weyler W, Sanford K: **The commercial production of chemicals using pathway engineering**. *BBA-Protein Struct M* 2000, **1543**:434–455.